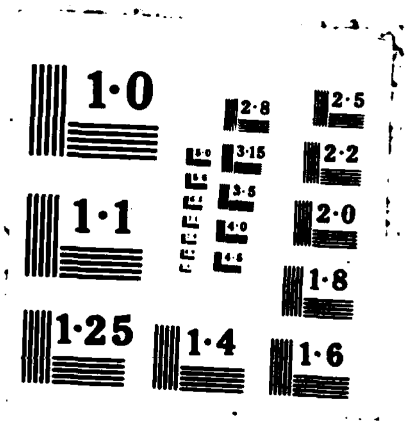
AD-A188 998 AUTOMATED INTERACTIVE SIMULATION MODEL (AISIM) VAX VERSION 50 TRAINING MA..(U) HUGHES AIRCRAFT CO FULLERTON CA GROUND SYSTEMS GROUP V ALLERTON ET AL. 29 MAY 87 HAC-1854896-2 ESD-TR-87-232 1/2

UNCLASSIFIED F/G 12/5 NL


1·0
2·8
2·5
3·15
2·2
3·5
1·1
2·0
4·0
1·8
1·25
1·4
1·6

ESD-TR-87-232 CDRL 106

Automated Interactive Simulation Model (AISIM) VAX Version 5.0 Training Manual

Hughes Aircraft Company
Ground Systems Group
P.O. Box 3310
Fullerton, CA 92634

VICKY ALLERTON
GLORIA BOICE
SUSAN SWEET

29 May 1987

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

Prepared For

ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
DEPUTY FOR DEVELOPMENT PLANS AND SUPPORT SYSTEMS
HANSCOM AIR FORCE BASE, MASSACHUSETTS 01731

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE

A188998

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION Unclassified	1b RESTRICTIVE MARKINGS BH7432
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for Public Release; Distribution Unlimited.
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S) CDRL 106 1854896-2	5 MONITORING ORGANIZATION REPORT NUMBER(S) ESD-TR-87-232
6a NAME OF PERFORMING ORGANIZATION Hughes Aircraft Company Ground Systems Group 6b OFFICE SYMBOL (If applicable)	7a NAME OF MONITORING ORGANIZATION Hq, Electronic Systems Division/XRSE
6c. ADDRESS (City, State, and ZIP Code) P.O. Box 3310 Fullerton, CA 92634	7b ADDRESS (City, State, and ZIP Code) Hanscom AFB Massachusetts, 01731-5000
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F19628-86-C-0070
8c. ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO / PROJECT NO / TASK NO / WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)
Automated Interactive Simulation Model (AISIM) VAX Version 5.0 Training Manual

12 PERSONAL AUTHOR(S)
Vicky Allerton, Gloria Boice, Susan Sweet

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Final	FROM 5/14/86 TO 5/15/87	1987 May 29	162

16 SUPPLEMENTARY NOTATION

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	AISIM, Simulation, Training, Models,

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

This document is the Training Manual for the Hughes developed Automated Interactive Simulation Model (AISIM). This manual provides step-by-step information necessary to begin using AISIM on a VAX 11/780 computer.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION Unclassified
22a NAME OF RESPONSIBLE INDIVIDUAL Michael F. Merriman, 1Lt USAF	22b TELEPHONE (Include Area Code) (617) 377-2716 / 22c OFFICE SYMBOL XRSE

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

CONTENTS

FIGURES

1. INTRODUCTION

This training manual presupposes virtually no programming experience and is intended to provide step by step information necessary to begin using AISIM version 5.0 as hosted on a VAX 11/780. It is not intended as a complete account of the system, and many topics covered in the companion AISIM User's Manual are covered here in less detail, or not at all. For further details on the operation of AISIM or on the kind of simulation AISIM is adapted to, the reader is referred to the more detailed AISIM User's Manual.

This manual consists of seven sections. This first section provides a brief overview of AISIM and its main concepts. Sections 2, 3 and 4 concern the Design User Interface (DUI), i.e., that part of AISIM in which models are created. Section 5 describes the complete construction of a simple model. Section 6 turns to the use of the Analysis User Interface—the part of AISIM where simulation and analysis occur--using the model developed in Section 5 as an example. Finally, Section 7 will document the creation, simulation and analysis of a more complex system.

1.1 MODELING

A computer model is a description of a system that is developed as a basis for calculations, predictions or further investigation. AISIM is especially designed to model systems that incorporate parallel processing. The purpose of an AISIM model is to give information on the workability of a system design, especially by providing statistics that serve to predict the operation of the modeled system if implemented.

Modeling is accomplished in AISIM by representing the elements of the system being modeled in terms of AISIM "entities." A detailed description of each entity is provided in Section 3 of the AISIM User's Manual. A general introduction to the types of system elements modeled by these AISIM entities is contained in section 1.3 of this manual.

1.2 OVERVIEW OF THE AISIM SYSTEM

AISIM consists of seven subsystems, each of which performs a distinct function. These subsystems are: (1) the Design User Interface (DUI); (2) the Analysis User Interface (AUI); (3) the Replot User Interface (RUI); (4) the Hardcopy User Interface (HUI); (5) the Library User Interface (LUI); (6) the File Management User Interface (FUI); and (7) the Help Editor Interface (HEI). Figure 1 shows each of these subsystems and the major components of each subsystem. The numbers above the boxes correspond to the numbering of the subsystems given in this paragraph. The words annotating each arrow are the commands used to invoke each subsystem or component, and the symbols in the boxes are the prompts seen after each subsystem or component has been invoked. Following figure 1 is a brief description of each of these subsystems.

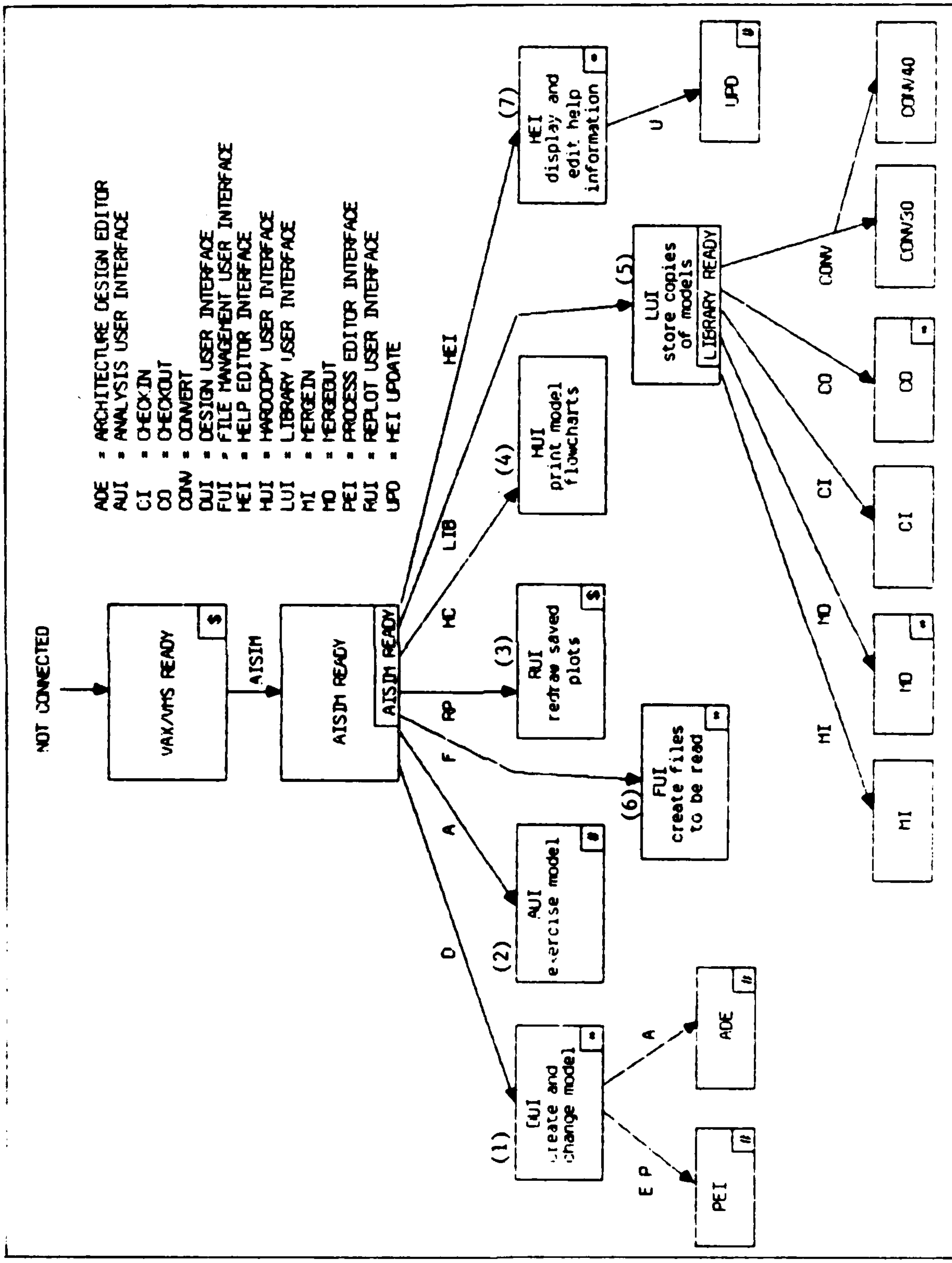


Figure 1. AISIM Levels of Operation

(1) DESIGN USER INTERFACE

The DUI is the facility which enables the user to create or alter models of systems. It contains two sublevels, the Architecture Design Editor (ADE) and the Process Editor Interface (PEI). The ADE is used to model the physical layout of the given system, which is called the architecture. The PEI is used to define the processes or logic that are associated with that architecture. Other model entities are defined at the DUI level.

(2) ANALYSIS USER INTERFACE

With the AUI one subjects the model defined in the DUI function to simulation runs that test the behavior and response of the modeled system to various hypothetical conditions. In this function statistics are gathered on the operation of the system during simulation and, if desired, graphs of selected parameters are generated and are available for plotting.

(3) REPLOT USER INTERFACE

The REPLOT function enables the user to plot the statistics from various executions of a model, and to combine these plots as desired for future reference.

(4) HCOPY USER INTERFACE

The Hardcopy function provides the connection between the AISIM system and a printing device for automatically producing hardcopies of model logic. Process flowcharts constructed in the DUI can be printed on an HP2631G printer/plotter, a TEK 4695 graphics copier, or the internal printer on the HP2623 terminal.

(5) LIBRARY USER INTERFACE

In the LUI the user is able to break apart and recombine parts of AISIM models, and obtain parts of models from a central system library. This feature is provided because some model components are used in other models and it is sometimes useful to store entire models for later reuse.

(6) FILE MANAGEMENT USER INTERFACE

The FUI function enables the user to create and manipulate external data files to be used by the READ Primitive, within an AISIM Process.

(7) HELP EDITOR INTERFACE

The HEI function provides a means for users to access and create help information.

1.3 OVERVIEW OF AISIM MODELING CONSTRUCTS

This section provides a brief description of AISIM modeling constructs, to be followed by a more precise description of them in subsequent sections.

With some qualifications, AISIM's modeling constructs can be divided into the following four categories: (1) those used to represent the operations, properties, structure and internal relations of the modeled system itself; (2) those used to represent the environmental stimuli to which the system model is exposed; (3) those which represent the physical layout of the system; and (4) those which represent and facilitate mathematical operations.

1.3.1 ENTITIES REPRESENTING ELEMENTS EXTERNAL TO THE MODELED SYSTEM

1.3.1.1 The Load Entity. The Load entity is used to represent aspects of the modeled system that trigger processes within it. The Load entity represents the normal demand which is placed on the modeled system. Loads are defined by specifying the nodes at which certain Processes are to take place within a given period (see Scenario), together with specifications of several parameters which indicate the schedule that the Process triggering follows. The definition of a Load will also assign a priority to each of the Processes being triggered.

1.3.1.2 The Scenario Entity. A Scenario is used to represent the external demand on a system (i.e., Process triggerings from the outside) throughout a simulation exercise. The Scenario divides a simulation run into a number of periods that determine the frequency with which Loads will be initiated. They will also trigger Processes in a way that is not systematically related to the Loads in order to represent abnormal impositions on the system.

1.3.2 ENTITIES REPRESENTING ELEMENTS INTERNAL TO THE MODELED SYSTEM

1.3.2.1 The Process Entity. A Process is used to represent the operations, decisions, actions or activities that can be decomposed and defined in terms of more fundamental AISIM entities, called Primitives. A Process can take place in one or more of the system's nodes (or may execute independent of the nodes) and can make use of one or more Resources.

1.3.2.1.1 The Process Primitives. Primitives, of which there are 28, are the elements of which Processes are composed. A Process may be considered to be a collection of Primitives whose sequential execution describes the logic of the Process.

The 28 Primitives can be arranged into eleven categories according to similarity of function. For the present, rather than give the meaning of each Primitive individually, it is sufficient to describe the categories and in a general way characterize the roles that members of each will play in the definition of a Process.

1. <u>INTERNAL PROCESS EXECUTION CONTROL</u>. The Primitives

COMPARE
BRANCH
ENTRY
PROB
LOOP

serve as a "framework" for Processes, enabling the Process to branch (either unconditionally or under certain conditions) to another portion of the Process, or to repeat certain segments of the Process a specified number of times.

2. <u>RESOURCE ALLOCATION</u>. As mentioned earlier, a Process frequently competes with other Processes for Resources. The Primitives

ALLOC
DEALLOC
RESET
TEST
LOCK
UNLOCK

govern the allocation of Resources among the various competing Processes.

3. <u>PROCESS EXECUTION CONTROL</u>. Since a principal feature of AISIM is its capacity to model parallel Processing, i.e., distinct Processes executing at the same time, these Primitives govern the timing of various Processes in the system relative to one another. The Primitives

CALL
SEND
SUSPEND
RESUME
WAIT

will either interrupt the Process in which they stand, or trigger or re-initiate some other Process.

4. <u>QUEUE HANDLING</u>. The Primitives

FILE
FIND
REMOVE

govern the placement and retrieval of Items in Queues that have been defined by the user.

5. <u>ITEM HANDLING</u>. The Primitives

CREATE
DESTROY

govern the introduction and elimination of a system's transient data elements.

6. VARIABLE MANIPULATION. The Primitive

ASSIGN

governs the assignment of values to variables or attributes (both numerical and non-numerical).

7. MATHEMATICAL OPERATIONS. The Primitive

EVAL

governs calculations, invoking standard mathematical functions and operations or making use of user-defined Tables.

8. TIME SEQUENCING. The Primitive

ACTION

which is associated with the Action entity described below, is included in Process definitions to indicate the time a certain Action (or process, decision, etc.) takes up without further describing the Action's nature.

9. DEBUGGING. The Primitive

TRACE

is not used to represent a system's operations, but is rather provided as a debugging facility to aid the user in the task of tracing a history of Process execution during simulations.

10. INPUT/OUTPUT. The Primitives

READ
WRITE

allow the input and output of AISIM variables (both numerical and non-numerical) from and to external files.

11. DOCUMENTATION. The Primitive

COMMENT

allows documentation to be entered into the AISIM model.

1.3.2.2 The Resource Entity. A Resource entity represents a component of the modeled system which may be necessary for the execution of a Process. Typically, a Resource will be required for more than one Process. Where several Processes demand a Resource that can serve only one Process at a time, all but one will stand in a queue until the Resource is available for them. The order in which the Processes will make use of the contended Resource is a function of the priorities associated with the various requests for the Resource.

1.3.2.3 The Action Entity. The Action entity is used in conjunction with the ACTION Primitive to represent any action, activity, decision, etc. that consumes time. The entity is automatically created by the system when an ACTION Primitive is placed.

1.3.2.4 The Legal Path Table. The Legal Path Table (LPT) is a set of routes or paths between nodes in the system's architecture. The LPT is selected from all the possible paths between the nodes along the links, so that there is but one permissible routing of communication between the various nodes in the architecture. The LPT is accessed by several other elements of AISIM such as the EVAL Primitive, the keywords $NODE, $NXTNODE, and $LINK, and the Message Routing Submodel Processes.

1.3.2.5 The Queue Entity. A Queue represents any holding area, such as a memory buffer or job queue, for elements waiting to take up their role in the operation of the system. User-defined Queues can be used as a holding area for Items. A user-defined Queue can be manipulated in a number of ways described later and in the AISIM User's Manual section 3.4.

1.3.2.6 The File Entity. The File entity is used to associate an external file with a model. This entity is used in conjunction with the READ and WRITE Primitives. The AISIM user does not have control over the creation and deletion of a file entity. It is automatically created when the first READ or WRITE Primitive accessing the file is placed and automatically deleted when the last READ or WRITE Primitive accessing the file is deleted.

Each File entity contains the logical name of a file to be accessed during the simulation. Associated with each logical file name, there must be an actual, physical file to which data is written or from which data is read. AISIM maintains a file called project.FNM that contains a list of all logical file names in a model and the physical files associated with each of those logical file names. During a Design session, AISIM reads in the information in the ".FNM" file for a model, makes updates as necessary as the user updates the model, and then rewrites the file at the end of the Design session.

1.3.3 ENTITIES WHICH SUPPORT MATHEMATICAL OPERATIONS

1.3.3.1 The Constant Entity. A Constant is an entity whose value does not change during a simulation run. Constants are specified or altered in the DUI and can be edited before a simulation run in the AUI but cannot be changed (and do not change) once the execution of a model has begun. Several parameters required in the definition of an AISIM model, (e.g., the number of Resource units available, the period length of the simulation and the size for Queues) can only take Constants or simple numerics as values.

1.3.3.2 The Variable Entity. Variables, by contrast, are entities whose values can change during the exercise of a model. The majority of the parameters in the specification of a model can take Variables as values.

1.3.3.3 The Table Entity. Tables are single-value, single-argument functions defined by the user. They may be defined either as discrete, continuous, or alphanumeric and may have from 1 to 15 entries. Tables are accessed by the EVAL Primitive and serve as a supplement to the mathematical operations automatically available as part of the EVAL Primitive.

1.4 OVERVIEW OF THE AISIM HELP FACILITY

The HELP Facility provides a means for the AISIM user to receive and disseminate information pertinent to the AISIM system. It allows users to display information on AISIM functions and modeling concepts or it allows them to enter and display user supplied messages.

1.4.1 DISPLAYING HELP. Entering the command HELP from any AISIM function command level will enable the user to request that help information be displayed. Users can either be guided through to the needed information by a series of prompts, or they can directly access the information by specifying the specific topic. The format of the display screen is consistent and shows the information in single screen pages. When a specific topic has more than one screen of information the user is prompted with the following message to continue viewing the subsequent screens:

PRESS RETURN TO CONTINUE

Upon completing the last screen of information the user sees:

NO FURTHER INFORMATION, PRESS RETURN TO CONTINUE HELP

Prior to terminating the initial request the user is given instructions on the additional help available.

1.4.2 CREATING HELP. The Help Editor Interface (HEI) UPDATE command is used to invoke the Update Function. This interface is used to create and edit user provided help information. There are three categories for the user information to be contained under: notes, guidelines, and procedures. The valid Update commands are: Add, Change, Delete, Help, List, Save, and End. Only one user at a time is allowed to use the Update Function in order to prevent multiple users from changing the help database simultaneously.

2. CREATING SYSTEM ARCHITECTURES

With the basic understanding of AISIM modeling concepts presented in the previous section, the reader should now be able to interact with the DUI. The exercises here are intended both to deepen the user's grasp of AISIM modeling constructs and to familiarize him with the prompts encountered while interacting with the DUI. In general, it is not a good idea to begin the design of an AISIM model without having done research and preparation on paper beforehand. However, as a teaching device, we shall develop fragments of an architecture from requirements formulated as we go along.

The method of logging on and invoking AISIM is computer-specific so we shall assume that the user has reached the point at which the computer prompts him with

```
AISIM READY
```

The user will have been offered a collection of information that looks something like that depicted in figure 2.

```
================================================================================

                    This is AISIM Production Version 5.0V
                                  5/15/87

 ***Please report any problems to:  The MITRE Corporation D-76, (617) 271-2274
================================================================================
```

Figure 2. Typical Display Upon Entering the AISIM READY Level.

To enter the DUI, type

```
design project(test) term(trmtyp)
```

"test" is the name of the model to be designed. Trmtyp represents the type of terminal being used.

The valid terminal types are the following:

```
HP   - HP2647A, HP2648A
HP23 - HP2623
VT   - VT100 with Selanar graphics
TEK  - Tektronix 4105
```

The user will be prompted with information that looks something like that shown in figure 3.

```
AISIM READY
design project(test) term(hp)
CURRENT PARAMETERS IN EFFECT:
VERSION:      PRODUCTION VERSION 5.0
TERMINAL:     HP
PROJECT:      TEST
USER:         [USER]
ENTER YES TO PROCEED, NO TO ABORT...
```

Figure 3. Typical Information on Entering the DUI

By typing

NO

the user will return to the AISIM READY level. Typing

YES

will put the user in the DUI and the screen will display an asterisk to indicate that the user may enter DUI commands.

Obtain help information on the DUI by typing

HELP

The following information will appear on the screen as shown in figure 4 as the user enters a carriage return in response to the prompt to continue.

```
FUNCTION LEVEL,DUI

                         DESIGN USER INTERFACE (DUI)

The DUI and its lower levels are used to define a model by creating,
modifying, or deleting AISIM model entities.  The Action, Constant, Item,
Load, Process, Queue, Resource, Scenario, Table, and Variable entities are
created and edited at the DUI level, using the EDIT command.  The descriptions
of File entities can be modified using the EDIT command.  The Process entities
which represent operations in the modeled system are created and edited at a
sublevel of the DUI level called the Process Editor Interface (PEI).  The PEI
is invoked by issuing the EDIT command (at the DUI level) and specifying a
Process as the entity to be edited.  A system architecture and its related
Legal Path Table, nodes, and links are defined in a second sublevel of the DUI
called the Architecture Design Editor (ADE).  The ADE is invoked by issuing
the ARCH command at the DUI level.

AISIM has a restricted character set for all data entered into a model.  Only
the characters A-Z, 0-9, "$" (dollar sign) and "_" (underscore) may be used
for model entity names and parameters.  Any time the user places an invalid
character in a name or a form field, the user will be requested to correct the

PRESS RETURN TO CONTINUE
```

Figure 4. DUI Help

```
FUNCTION LEVEL,DUI

invalid character.  Any printable characters are allowed in "comment" and
"description" fields of forms and in user-added help text (see section 11.2).

When creating and editing entities in the DUI level, the system prompts
the user for further information by use of forms.  Each form specifies the
required and optional attributes of its respective entity-type.  The areas
on which information is to be entered appear in "reverse video" (dark
characters on a light background), and indicate the attributes that are to
be supplied by the user.

Each time the user presses the keyboard carriage return key, the character
cursor is positioned to the start of another designated area.  The user
enters parameters requested by the form by keying in the desired
alphanumeric information.  If the user changes his mind about the
parameters previously keyed in, he may alter them by merely writing over
the old information.  When the user is satisfied with the contents of the
form, he inputs it to the computer by exiting the form.  Below is a
complete description of the use of forms.

While the user is in the DUI, all changes are made to a working copy of
PRESS RETURN TO CONTINUE
```

Figure 4. DUI Help (continued)

```
FUNCTION LEVEL,DUI

the user's database.  When the user issues a SAVE command during or at the
end of the DUI session, the working database is copied back into the
user's real database.  This procedure enables the user to change his/her
mind about changes made in the working database and to protect the user's
real database in case the computer crashes during a DUI session.

AVAILABLE SUBTOPICS ARE:
ARCH              COPY              DELETE            EDIT
END               HELP              LIST              SAVE
UNITS
SUBTOPIC NAME?
```

Figure 4. DUI Help (continued)

Entering ARCH in response to the Subtopic Name prompt will display the information in figure 5.

```
FUNCTION LEVEL,DUI,ARCH

The ARCH command is used to invoke the Architecture Design Editor (ADE).

This command is valid only in the DUI Ready Level.

COMMAND SYNTAX:

     ARCH

     A

FUNCTION RESULT:

The ADE is invoked so that the architecture is built under the project
designated by the DESIGN command.  A # prompt is provided for the user to
input ADE commands.  These commands are discussed in section 6.3.
NO FURTHER INFORMATION, PRESS RETURN TO CONTINUE HELP
```

Figure 5. ARCH Help

By entering the carriage return mentioned in figure 5 a screen will be displayed providing an opportunity to request additional help. Entering a carriage return will return the user to the DUI.

When the computer redisplays the prompt "*", enter the Architecture Design Editor (ADE) by typing

ARCH

A grid like that in figure 6 will appear on the screen.

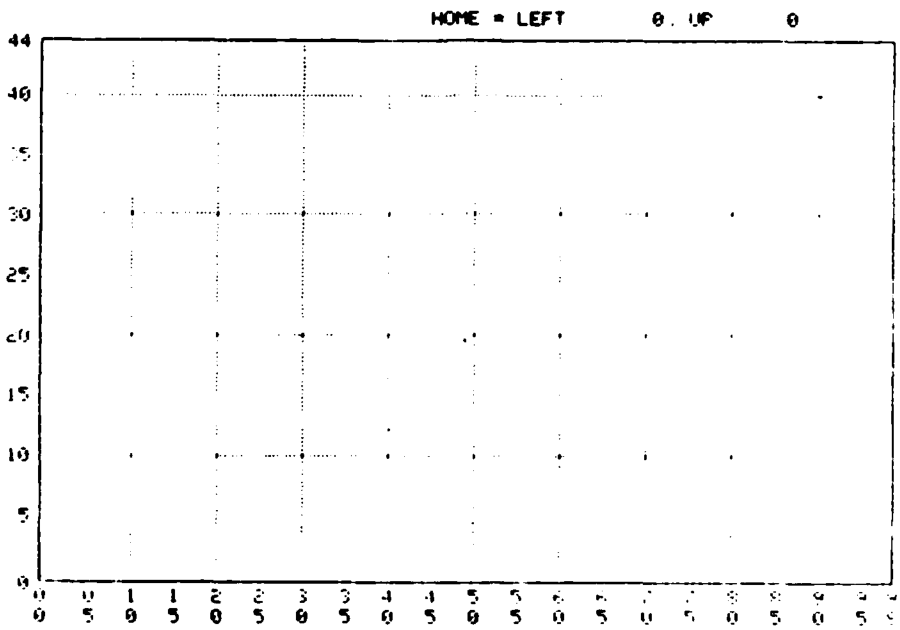


Figure 6. Grid on Which an Architecture is Designed

The AISIM constructs manipulated in the ADE are nodes which represent the hardware elements of a system and links which represent lines of communication between them. The physical layout of the system is represented by placing nodes and links on the grid to represent various hardware elements of a system and their (available) lines of communication. A Resource modeling entity is automatically created for each node or link when it is placed in the architecture.

As a mnemonic aid in distinguishing system elements, AISIM provides fourteen geometrical symbols for nodes. The symbols are called by the three-letter designations given in figure 7.

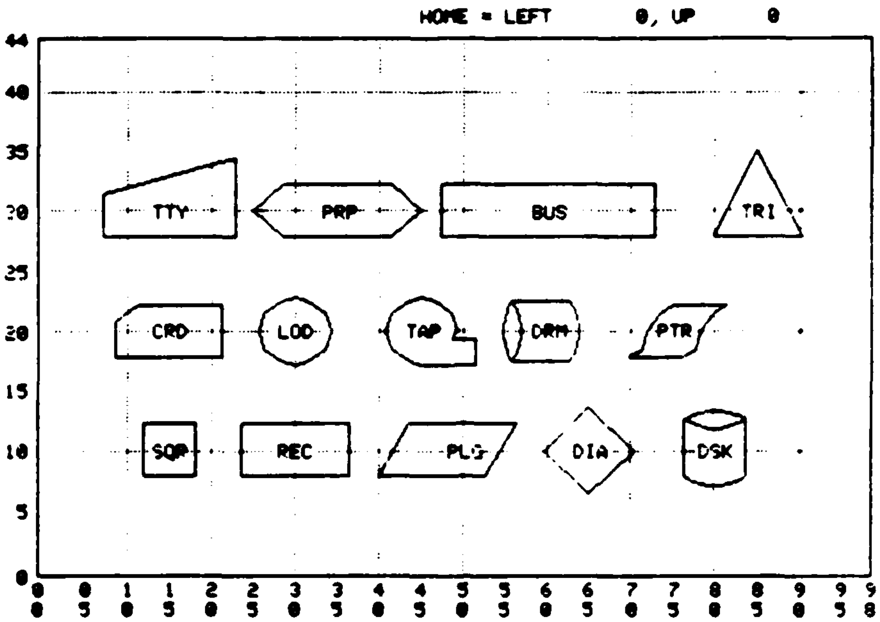


Figure 7. Designations of the Fourteen Symbols

With two exceptions these node symbols differ from one another only in their appearance. The two exceptions are the so-called "leaf-nodes" TTY and LOD. These nodes may be connected to the modeled architecture by only one link. All other nodes may be connected to any number of other nodes through any number of links. The rationale for this restriction is explained in the AISIM User's Manual, section 6.3.3.

2.1 DRAW AND NODRAW MODES

In the ADE there are two modes called DRAW mode and NODRAW mode. When the user is in DRAW mode, all architecture commands which change something in the architecture display cause the display to be updated immediately. If the user is in NODRAW mode, the architecture display is not automatically updated. The user can cause the display to be updated by typing

REDRAW

DRAW mode is the default on HP terminals and TEK4105 terminals. NODRAW mode is the only mode available on VT100 terminals. This restriction is due to a lack of capability in the terminal to make it operate like the other terminals. Therefore, if a user is on a VT100, he must type REDRAW to see the results of any ADE commands. The following discussions assume the user is on a terminal other than a VT100 except where specifically noted. If the user is on a VT100 terminal, he will need to use the REDRAW command to view the results of the ADE commands described in the following sections.

2.2 DEFINING ATTRIBUTES FOR SYMBOLS

As mentioned earlier, when a symbol is placed in the architecture, an AISIM entity called a Resource is created to represent the hardware element depicted by the node or link. Resources can have a number of user-named attributes. The DEFINE SYMBOL command allows the user to associate attributes with each symbol type so that when placed in the architecture, the Resource will be created with all attributes defined. For example, after typing:

DEFINE SQR

(SQR could be replaced with any of the 14 symbol mnemonics or the mnemonic CON if a link is being defined.) The user will be prompted by a form as shown in figure 8.

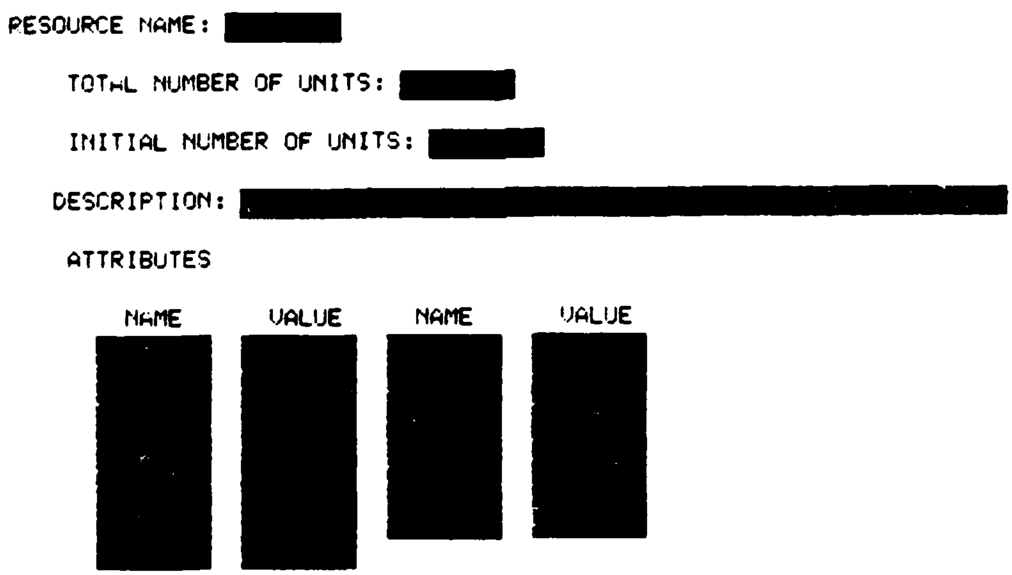


Figure 8. Symbol Definition Form

The user can add or modify fields within this form by using the terminal keys as defined in appendix A. Any values in the inverse video fields of the form are default values supplied from the AISIM design data base. The user may change these fields by typing over the existing values. The user may enter up to fifteen attribute names and related values of his choice into the attribute fields. For example, when defining attributes of a symbol type which is to represent a disk in the modeled system, the attribute names may be something like seek time, latency, etc., and the values would be the corresponding values for the particular disk being modeled.

A second use of the DEFINE SYMBOL command is the following:

DEFINE SYMBOL,RESOURCE NAME

where SYMBOL is one of the symbol mnemonics or CON, and RESOURCE NAME is the name of an existing Resource entity. If the named Resource entity exists, a form similar to the form shown above would be displayed. Instead of the default attributes, the form displayed would have the names and values of any attributes previously defined for the Resource entity referenced in the command.

This command will only be accepted if a Resource entity has been previously defined before entering the ADE. Since the user has not defined any Resource entities in his test data base yet, this command would fail. The user might want to try this command later.

2.3 PLACING NODES ON THE GRID

To place a node at a certain location on the grid--i.e., centered on that location--issue the PLACE command designating (1) the type of node to be placed, (2) a user-given name, and (3) horizontal and vertical position coordinates. One can also opt to indicate the size of the geometrical shape if the default value, equal to the number of characters in the user-given name, is unsuitable. To center a square named NODE1 twenty units from the left-hand side and thirty units from the bottom in figure 7 above, type

PLACE SQR,NODE1,20,30

Figure 9 shows the screen display that would result from this command.

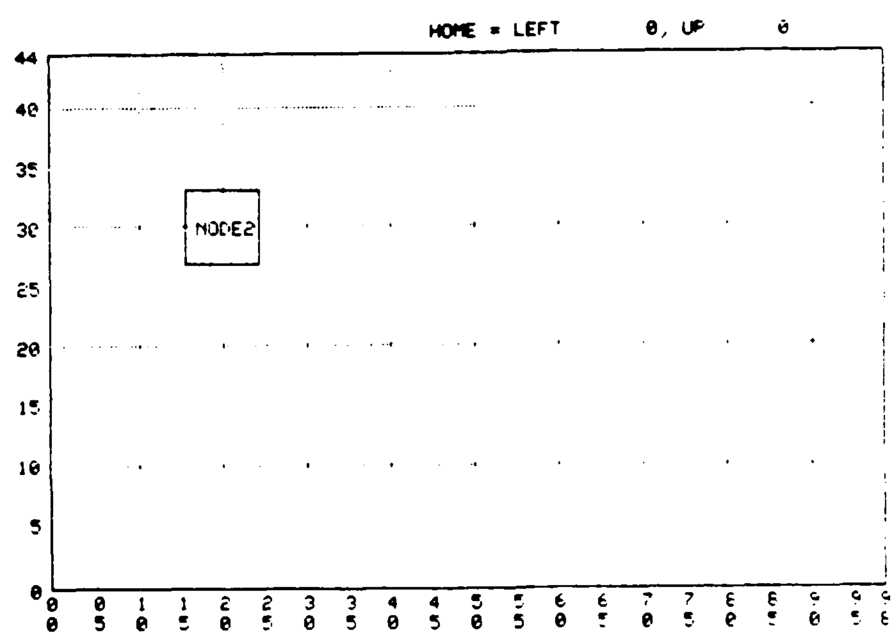


Figure 9. Architectural Grid With a Single Node

All nodes are placed in this way. To place nodes in the positions shown in figure 10, type the following sequence of commands ("P" is an abbreviation for "PLACE"):

```
P TTY,NODE2,15,10

P PRP,NODE3,45,30

P TRI,NODE4,70,10

P TAP,NODE5,45,15

P CRD,NODE6,75,35
```

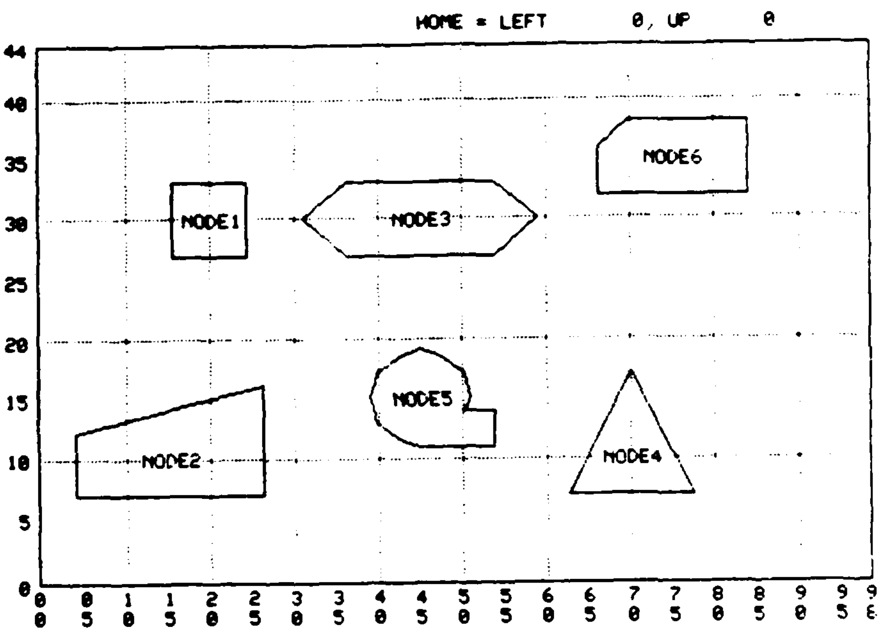


Figure 10. Six Nodes on an Architectural Grid

2.4 CONNECTING NODES

The second step in creating a system architecture is the placement of connections between the nodes. The CONNECT command is used to connect two nodes. Such connections, or "links", are defined by specifying (1) the node from which the link is to run, (2) the node to which the link is to run, and (3) a user-given name of the link. To place a link called "LINK1" from NODE1 to NODE2, type

```
CON NODE1,NODE2,LINK1
```

This command places a cursor at NODE1 if the user is on an HP terminal, or in the lower left corner if the user is on a TEK terminal; typing any alphanumeric character other than a period causes a straight line to be drawn between the centers of the two nodes, thereby drawing the link. The graphic result is shown in figure 11.

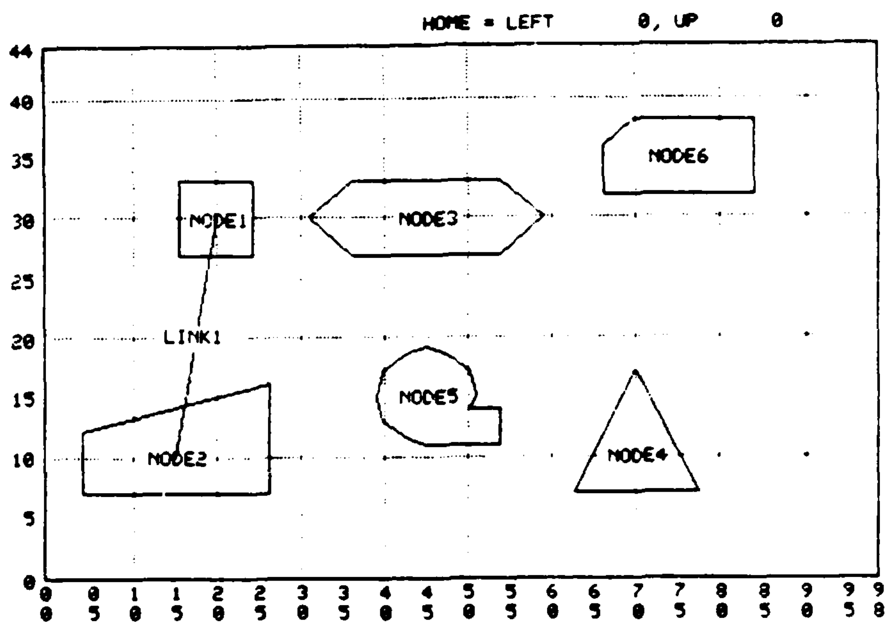


Figure 11. Architecture with One Link Defined

Links need not always appear as straight lines, as is shown in figure 12.

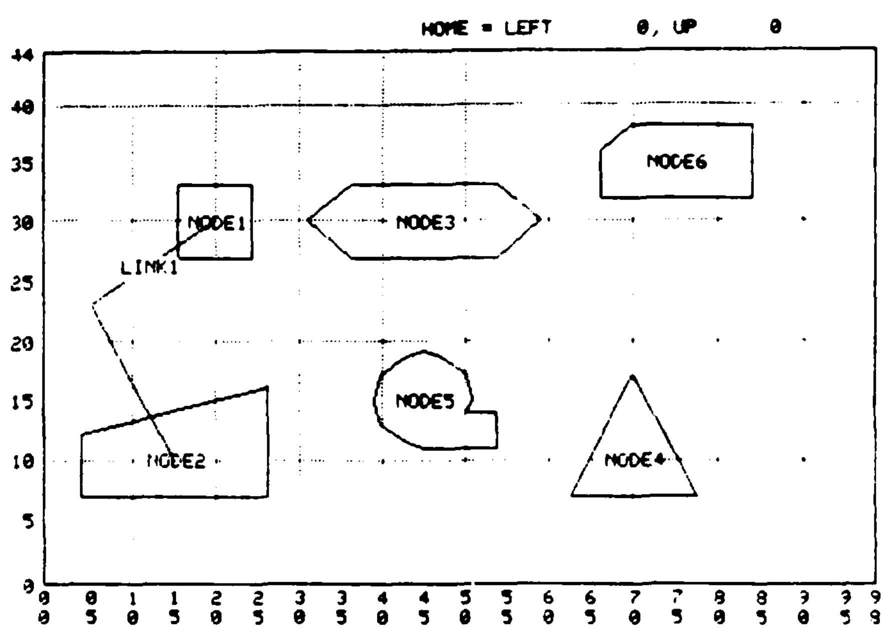


Figure 12. Architecture with a Bent Link

To create links that bend, enter the CON command as above. Then using the graphics cursor controls on the terminal (either the arrow keys on the HP terminals or the joystick button on the TEK terminal), the cursor can be moved to the spot where the link is to bend. When the cursor is at the point of the bend, type in a period (.). If no further bending is desired, typing any other non-period alphanumeric character will complete the connection. The resulting connection will resemble the one depicted above in figure 12.

Links may be given more than one bend by repeating the sequence of moving the cursor and typing a period (.), and then depressing any non-period character only when all the desired bends (up to 5 bends, i.e., six segments) have been created.

To create the links shown in Figure 13, type the following sequence of commands:

```
CON NODE1,NODE4,LINK4

CON NODE3,NODE6,LINK3

CON NODE3,NODE5,LINK5
```

NOTE: If the user is logged on through a VT100 terminal, there is no capability to create bent links. All connections are automatically made as straight lines between the two nodes.

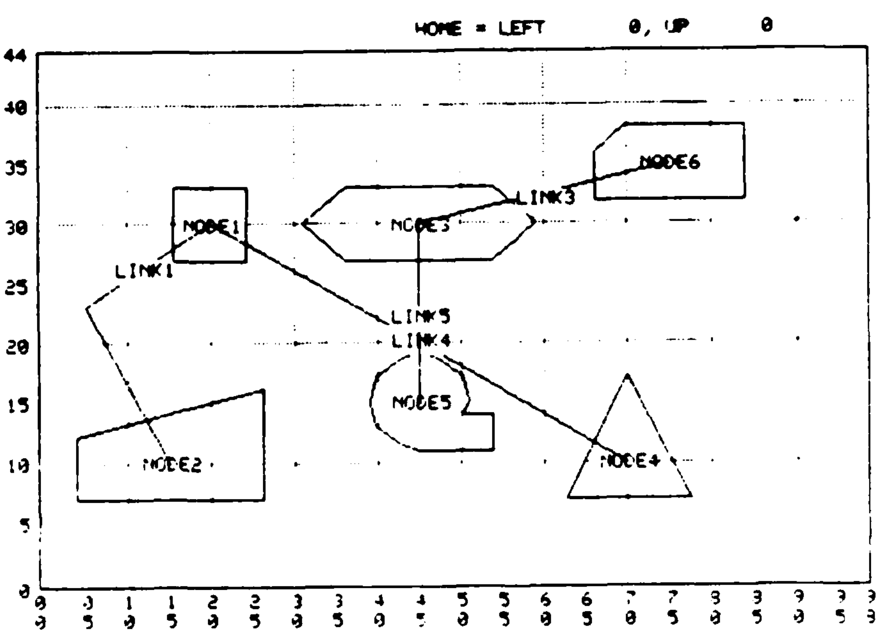


Figure 13. Architecture with Four Links

2.5 CHANGING THE SIZE, TYPE, AND NAME OF NODES AND LINKS

The size, type, or name of nodes and the names of links can be changed using the CHANGE command. By typing the following commands, nodes and links may be altered:

```
CHG NAME,NODE1,NODEX

CHG TYPE,NODE2,LOD

CHG SIZE,NODE3,7

CHG NAME,LINK4,LINKZ
```

The user may note the changes on his screen. By typing the following commands, the architecture is returned to its original configuration:

```
CHG NAME,NODEX,NODE1

CHG TYPE,NODE2,TTY

CHG SIZE,NODE3,5

CHG NAME,LINKZ,LINK4
```

As mentioned earlier, Resource entities are created by the AISIM system to model the architecture elements. When the name or type of a node is changed or the name of a link is changed, the appropriate changes are also made to the associated Resource entities. That is, when a node or link name is changed, the associated Resource name is changed. When the type of a node is changed, new attributes may replace the existing attributes of the Resource since different attributes may be defined for the new symbol type. Refer to Section 2.2 of this manual.

2.6 DELETING NODES AND LINKS

Existing nodes and links may be deleted from a system architecture with the DELETE command. For this example, to eliminate the connection between NODE1 and NODE2 type

```
DELETE LINK1
```

The result on the screen would be that the link named "LINK1" would disappear.

When a node is deleted, all of the links associated with it also disappear. As an example type

```
DELETE NODE6
```

The result of deleting LINK1 and NODE6 is shown in figure 14. Note that LINK3 disappeared also.

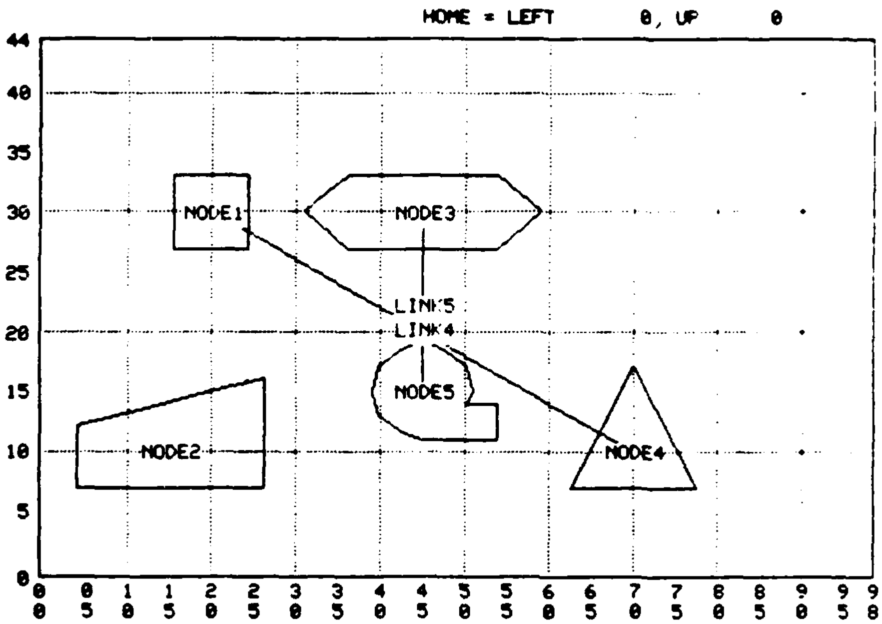


Figure 14. The Result of Deleting LINK1 and NODE6

2.7 MOVING PREVIOUSLY PLACED NODES

The location of a node on the architecture grid may be changed with the MOVE command. For example, to move NODE4 from its current position to the coordinates 55,5 one issues the command:

```
MOVE NODE4,55,5
```

The graphic result is shown in figure 15. The symbol is now centered at 55,5 with all of its previously defined connections intact.

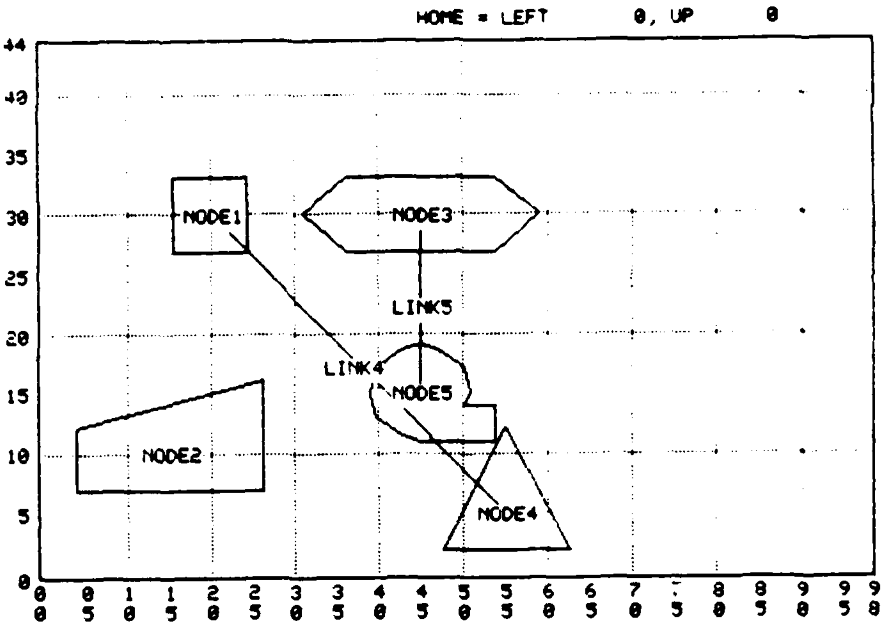


Figure 15. The Result of Moving NODE4

2.8 RECONNECTING EXISTING LINKS

The previous example of moving NODE4 created a problem that can be solved with the command RECON. As figure 15 shows, the link between NODE1 and NODE4 now runs through NODE5. The connection can be made to bend around NODE5 by first typing

RECON LINK4

This command will delete the existing graphics for the link between NODE1 and NODE4 and, as in the original CONNECT command, place the cursor at NODE1. Using the sequence of cursor movements and periods (.) described in section 2.4, up to five bends in the existing connection between NODE1 and NODE4 can be created. To complete the connection, type any non-period character. The graphic result will be something like that shown in figure 16.

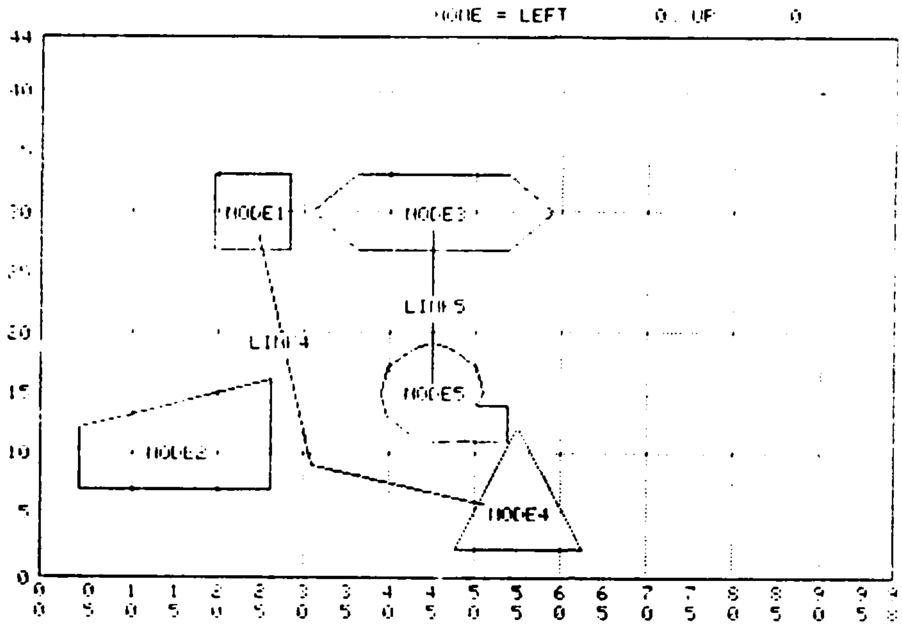


Figure 16. Architecture After Reconnecting

NOTE: The same restriction applies to the RECON command as the CONNECT command when the user is logged on to a VT100 terminal; i.e., only one-segment connections are allowed (see section 2.4).

2.9 ALTERING ONE'S VIEW OF THE ARCHITECTURE GRID

The usable grid space in ADE is actually larger than what can be displayed on the terminal screen at one time. If an architectural design is too large for the screen to accommodate, different parts of the total workspace can be viewed and manipulated through the WINDOW command. The WINDOW command allows the directional change of the user's view of the grid. The command specifies the direction of change--up, down, right or left--as well as the number of grid units the view is to be changed.

For example, to move the view of the screen in figure 16 down 15 units, type

WINDOW D,15

Figure 17 shows the result of this command.

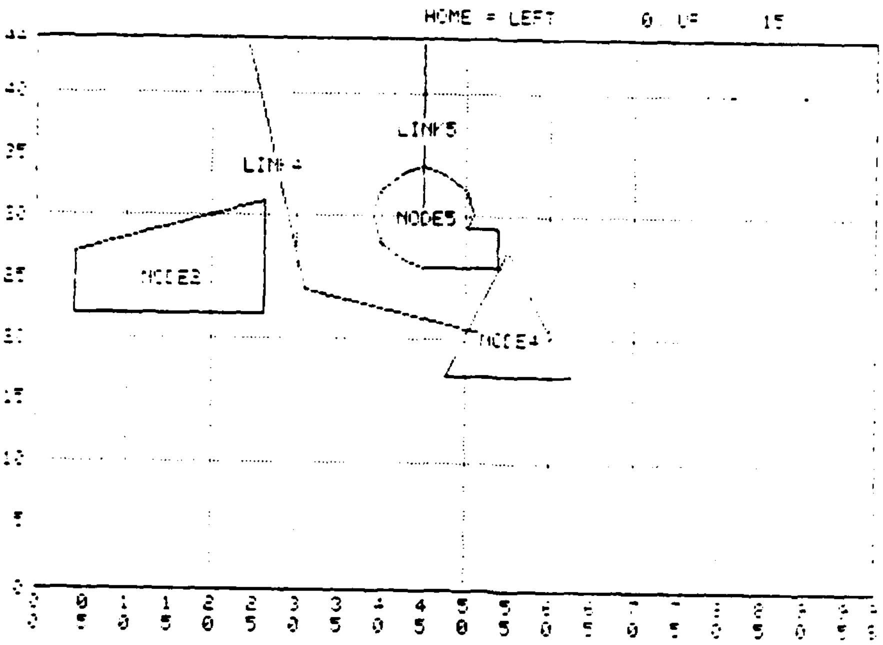


Figure 17. Result of WINDOW Command

The WINDOW command will accomplish both horizontal and vertical movements at the same time. To move our view of the screen further down 15 units and 20 units to the right, type

WINDOW D,15,R,15

Figure 18 shows the result of this command.

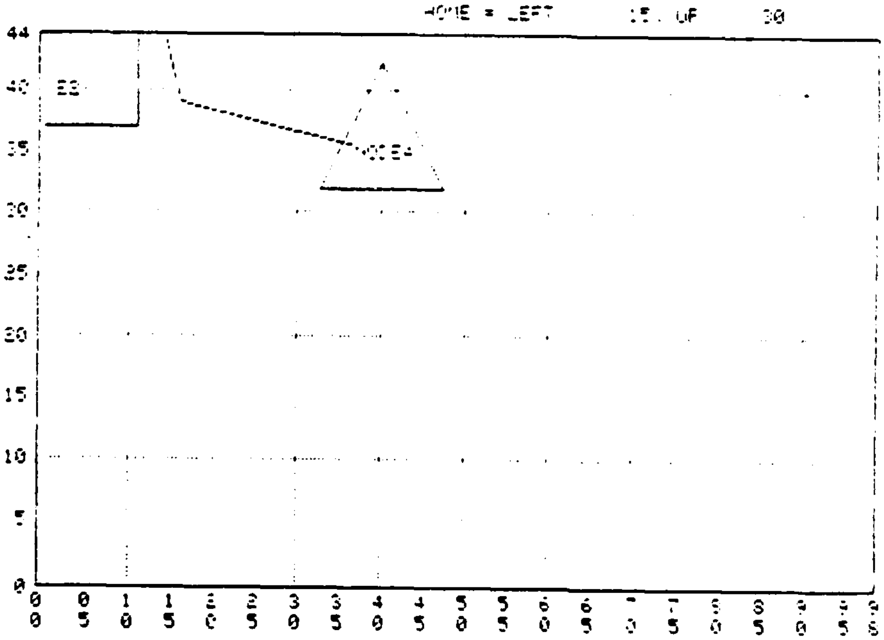


Figure 18. The Result of Further Use of WINDOW

Note that the WINDOW command parameters required to get back to the original (HOME) position are always displayed above the upper right corner of the architecture grid.

2.10 DEFINING LEGAL PATHS

The purpose of the Architecture facility is to specify routes of communication between hardware elements so that Process execution will be realistically related to the physical layout of a system. Such routes are represented by a Legal Path Table which specifies the links and the nodes through which communication from one node to another must take place. There are several methods of defining a Legal Path Table (LPT). Three methods are offered to the user at the end of an ADE session. These methods are predefined algorithms for the definition of an LPT which can be executed optionally at the user's discretion. See the AISIM User's Manual section 6.3.19 for details of how these algorithms function. For many architectures it is more economical to create the Legal Path Table while defining the configuration of hardware elements rather than using the algorithms mentioned above. If an LPT is generated according to the following discussion, the predefined algorithms should be bypassed since they would erase the LPT so defined.

Suppose we augment the architecture developed above with more links so that it resembles that shown in figure 19.

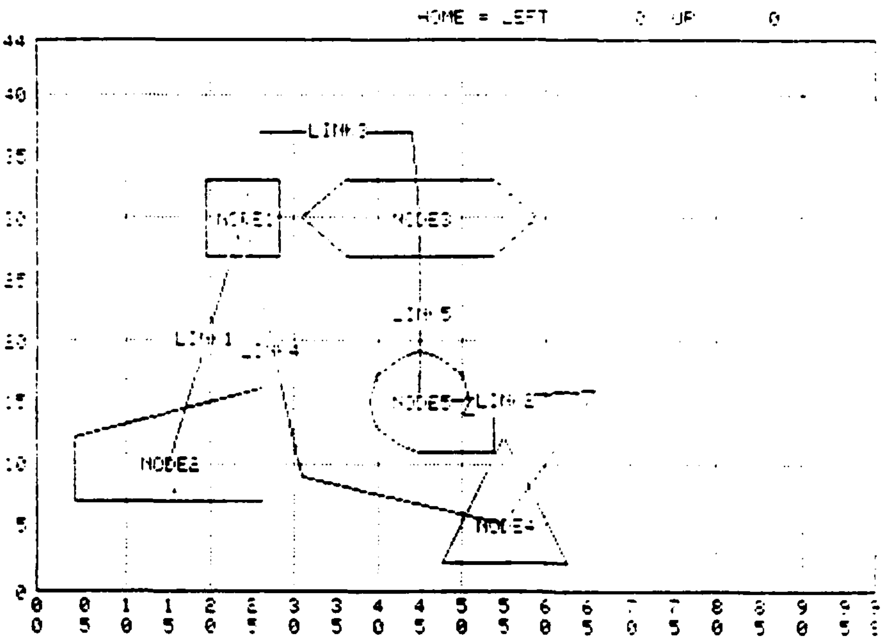


Figure 19. Augmented Architecture

The LPT is defined by means of the command DEFINE PATH. If, for example, NODE1 is to communicate with NODE4 along the communication lines represented by the links LINK3, LINK5, and LINK2, type

```
DEFINE PATH,NODE1,NODE4,LINK3,LINK5,LINK2
```

No confirmation will be displayed immediately at the screen, but the Legal Path Table will have been augmented to reflect the new paths. However, the command LIST PATH enables the user to inspect the Legal Path definitions currently in effect. To obtain a listing at the screen of the legal path from NODE1 to NODE4, type

```
LIST PATH,NODE1,NODE4
```

The resulting list is shown in the upper right-hand corner of figure 20.

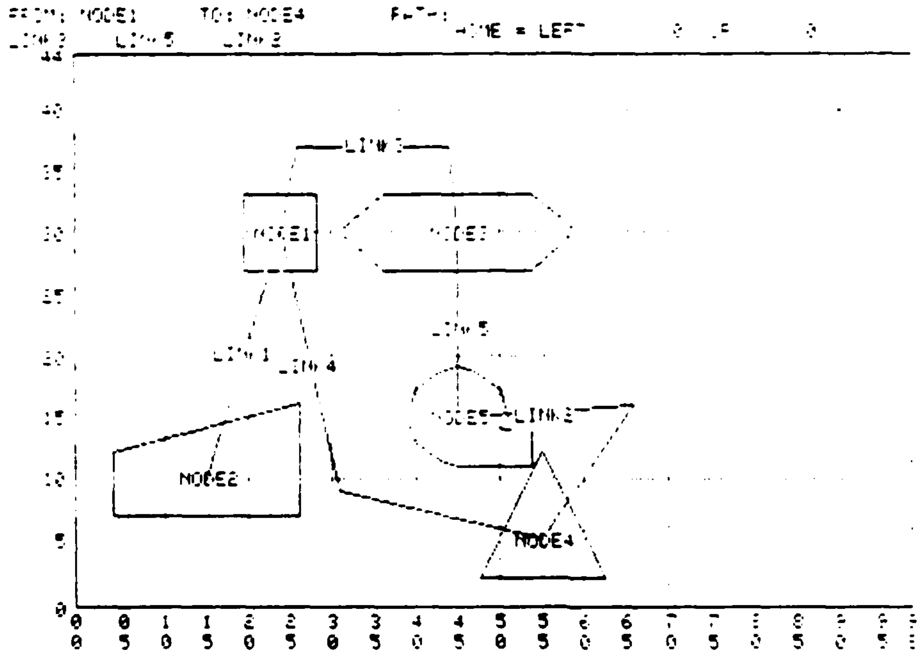


Figure 20. Typical List of Legal Paths Obtained in ADE

Note that paths from NODE3 and NODE5 to NODE4 have been automatically defined by the preceding DEFINE PATH command. The principle is that any time a legal path is defined through a number of nodes, the AISIM system creates legal paths from all nodes through which the path passes to the destination to node. Care should be taken in defining subsequent legal paths according to this method. Any conflicts of path routing in paths defined later would result in the elimination of previously defined paths. Following is an illustration of this operation of the system. Assume that the path has been created as above. If the user should now enter the command:

```
DEFINE PATH,NODE2,NODE4,LINK1,LINK4
```

not only would the path from NODE2 to NODE4 be established, but the path from NODE1 to NODE4 would be altered to be the direct path via LINK4. The paths defined automatically from the previous command (i.e., the paths from nodes NODE3 and NODE5 to NODE4) would still exist since there was no conflict with these paths and the newly defined path.

3. DEFINING PROCESSES IN THE DUI

Whereas the Architecture Design Editor (ADE) is used to represent the physical layout of a system, the Process Editor Interface (PEI) is used to represent the logic and data-handling behavior of processes in the system.

This section provides examples to familiarize the user with the commands and prompts used in the PEI. Earlier the user was urged to begin the design of an AISIM model with sufficient research and planning to fully understand the system to be modeled. However, as a teaching device, we shall develop fragments of a Process from requirements formulated as we go along.

The exercises here are intended both to deepen the user's grasp of the Process Primitives and to familiarize him with the prompts encountered while interacting with the PEI.

Assuming the user has just completed the foregoing section, entered an END command to exit the ADE sublevel, and another END command to bypass the LPT generation, he will be at the DUI level of operation. A "*" should be displayed. To invoke the PEI sublevel of the DUI, one enters the EDIT PROCESS command designating the name of the Process to be edited. Once in the PEI, the user can terminate the PEI session by entering an END command.

Consider first the simplest Process that could be of any use to the modeler of a system: the Process starts, a certain amount of time is taken with an Action and then the Process ends. This Process will be represented in AISIM by the START symbol, the ACTION Primitive and the END symbol. To represent such a Process in AISIM, one begins by issuing the command

```
EDIT PROCESS,EXAMPLE,NEW
```

which informs the system that one wishes to create a Process named "EXAMPLE". To alter a Process that has been previously defined, one would not enter the "NEW" part of the command. The computer will respond with a form to be filled in at the terminal. This is done by typing into the fields provided in the form. The form is shown in figure 21.

START

PROCESS NAME   NODE

ATTRIBUTES ATTACHED (YES OR NO)

PROCESS DESCRIPTION

START BLOCK TYPE

ENTER "PARM" FOR PARAMETER PASSING
ENTER "ITEM" FOR ITEM PASSING
ENTER "STD " FOR STANDARD PROCESS

Figure 21. Initial Form for Process

The NODE field asks for the node in the architecture with which the Process is associated. Since this Process is not yet related to an architecture, the field is left blank. The next field allows the assignment of attributes to the Process. For the present, we shall decline to do so. The field labeled PROCESS DESCRIPTION is for a comment to describe the Process. In this case, type "Example Process".

Depress the key that enters the form (see appendix A for specific key). The screen will go blank for a moment and then display the image depicted in figure 22.

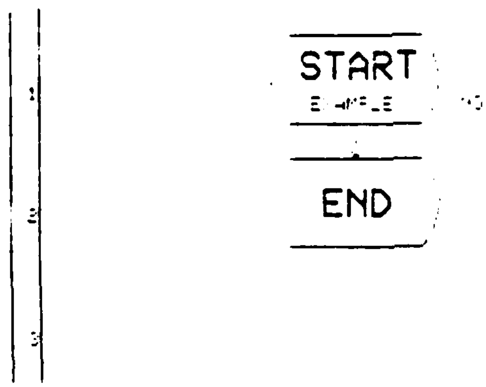


Figure 22. Graphic Display That Follows Entering a Standard Process

Much of the information you entered on the form now appears in this graphic representation of EXAMPLE. The "NO" to the right of the START symbol indicates the the Process has no attached attributes.

3.1 PROCESS EDITOR DRAW/NODRAW MODES

The Process editor maintains DRAW and NODRAW modes similar to the Architecture Design Editor except that both DRAW and NODRAW modes are available for all terminal types in the Process Editor. If the user is in DRAW mode, changes made to any Primitives will cause the specifically affected Primitives to be redrawn, or the entire screen will be redrawn if the affected Primitive is off the screen. If the entire screen is redrawn, the changed Primitive is placed at screen position three. If the user is in NODRAW mode, then any changes to Primitives will not cause the screen to be updated until a REDRAW command is entered, which will cause the display containing the most recently changed Primitive to be redrawn. The user can also use the commands TOP, BOTTOM, UP and DOWN to cause the display to be redrawn at a different location.

The default mode for the Process Editor is DRAW mode. The user can switch to NODRAW mode or back to DRAW mode by entering the command NODRAW or DRAW. The following discussions assume the user in DRAW mode.

3.2 ACTION

To place an ACTION Primitive between the Start and the End symbols, enter the command

P ACTION

which tells the computer to place an ACTION Primitive between the last Primitive defined and the END symbol. The computer will now display a new form to be filled in. This is shown in figure 23.

```
PARAMETERS FOR ACTION

  ACTION NAME:          OPTION: RESUME    METHOD:

  MEAN TIME:          DELTA TIME:          UNITS: SECONDS

  COMMENT:
```

Figure 23. Form for the ACTION Primitive

The field ACTION NAME requests a name for the Primitive in the Process, which should be identical with the name of the associated Action entity. (If it does not exist, the ACTION entity will be created along with the ACTION Primitive. Action entities are described in section 4.1.) We shall call it "Delay". The field COMMENT is a place to write a short reminder of what the ACTION Primitive is supposed to represent. The three fields METHOD, MEAN TIME, and DELTA-TIME enable the user to vary the time taken up by the ACTION by invoking various statistical distribution methods (such as exponent, uniform, etc. See section 3.9.1 of the AISIM User's Manual for a description of the valid distributions.) Assume that the ACTION always requires the same amount of time, and hence the MEAN TIME requested will be equal to the duration of the ACTION. Indicate the time with a variable whose value for this example is specified elsewhere, calling it "T1". The field OPTION specifies that the ACTION will be resumed, rather than restarted if it is interrupted. The field UNITS states that the unit of time is seconds.

The form should then be filled in thus: call the ACTION "Delay", set the method at "Constant", and set the MEAN TIME at "T1". Type the comment field "Action which causes delay". Leave the field labeled DELTA blank and let the OPTION and UNITS fields default to the displayed values. After leaving the form, the screen will display a new version of EXAMPLE, as depicted in figure 24.

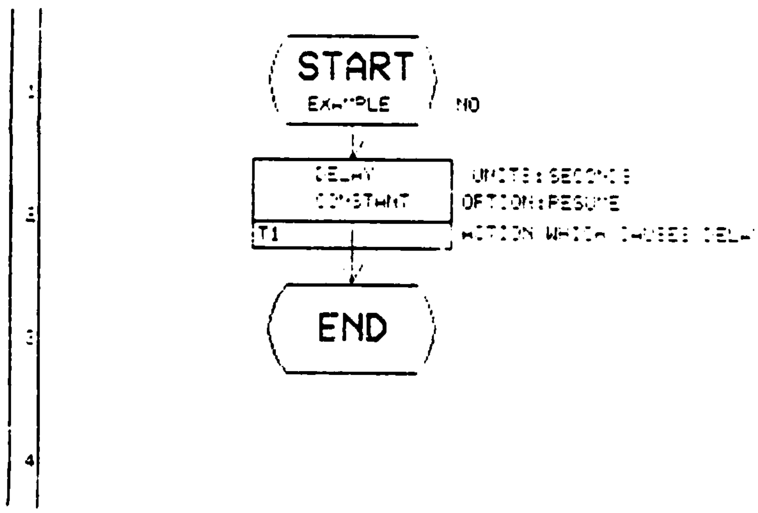


Figure 24. Process with a Single ACTION

3.3 HOLD

EXAMPLE may be augmented in a number of ways. For example, the ACTION Delay may be repeated in succession. There are two ways to repeat this ACTION in a revised version of EXAMPLE. First, one can place more copies of Delay in the Process, one after another, with the command

P ACTION

This command may be repeated as many times as one wishes the Action to be performed in the Process. The second method of creating several instances of an ACTION which is less time-consuming, involves the HOLD storage area. HOLD constitutes a storage area for Primitives that are likely to be used more than once with little or no alteration. To place a previously defined Primitive into HOLD, type

HOLD 2

The number 2 represents the position in the Process of the Primitive to be stored in HOLD. The position is indicated by the numbers in the column on the left. Hereafter, the ACTION Primitive "Delay" may be placed in a Process by typing

P HOLD

Each time this latter command is issued, the user will be presented with the form associated with the Primitive in case any small alterations in its parameters are to be made. Whether or not any alterations are made, leaving the form will result in the placement of the Primitive stored in HOLD in the Process being edited.

Figure 25 shows the display that will appear after several identical ACTION Primitives have been placed in succession or after the HOLD storage area containing the ACTION "Delay" has been placed. The procedure of repetition will occur as many times as the user requests it.

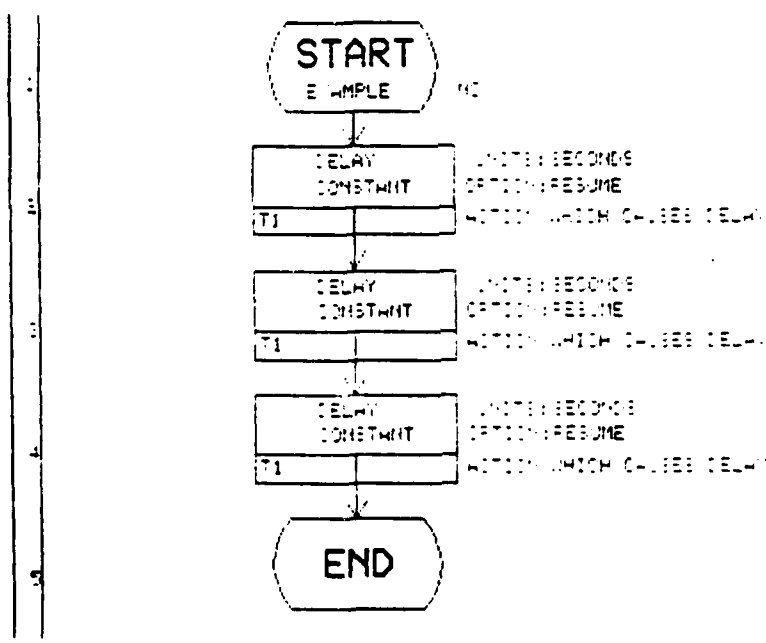


Figure 25. Process with Three Identical ACTION Primitives

3.4 ENTRY AND LOOP

If the ACTION "Delay" is to be performed a certain n number of times, as in the most recent version of EXAMPLE, a much simpler procedure is available than that of placing n instances of the ACTION between the START and END symbols. One can instead indicate more directly that a certain part of a Process is to repeat itself an n number of times. This is accomplished by means of the Primitives LOOP and ENTRY. Figure 26 shows EXAMPLE altered with LOOP and ENTRY Primitives to cause a triple repetition of the ACTION "Delay".

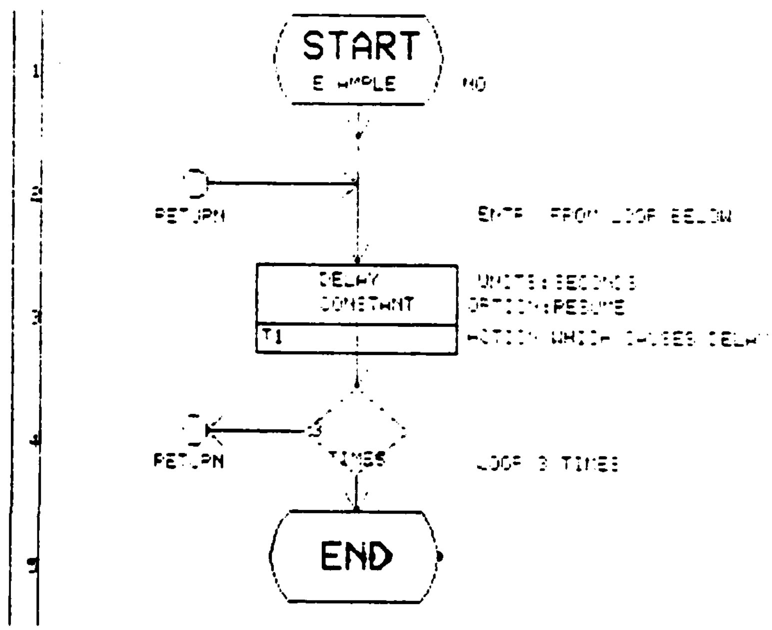


Figure 26. Process with Triple Repetition of the ACTION Delay

The diamond-shaped figure indicates that the line of processing is to be diverted to the point labled "Return" above it (it could have been given any label whatever up to 8 characters).

To effect the LOOP Primitive as shown in figure 26, we must first get rid of the two extra ACTION Primitives. This is done by typing the following commands:

DELETE 3

DELETE 3

P LOOP

The screen will show the form shown in figure 27.

PARAMETERS FOR LOOP:

LOOP TO LABEL:

LOOP TIMES

COMMENT:

Figure 27. Form for the LOOP Primitive

The first field asks for the name of the entry point to which Process control is to be diverted. The second asks for the number of times the primitives between the ENTRY and the LOOP are to be performed (i.e., control will be diverted to the entry point one less time than the number since the steps will have been executed once already when the LOOP is reached). The remaining field is self-explanatory.

The ENTRY Primitive must now be placed above the ACTION Primitive. Since the PLACE (or "P") command by default inserts the new Primitive just before the END symbol, a modified PLACE command is required to place a Primitive elsewhere in the sequence. To use this command, type

```
P ENTRY,2
```

The number "2" indicates where the Primitive is to be placed, in reference to the column of numbers on the left of the Process diagram.

The screen will then display the form shown in figure 28.

PARAMETERS FOR ENTRY:

ENTRY LABEL:

COMMENT:

Figure 28. Form for the ENTRY Primitive

The label will, in this case, be determined by the LOOP label previously defined. The ENTRY LABEL field should be entered exactly as in the LOOP LABEL, i.e., as "Return". Type an appropriate comment in the field provided, such as, "Entry From Loop Below". The result should be as in figure 26.

3.5 PROB, TEST, COMPARE AND BRANCH

Four other Primitives, PROB, COMPARE, BRANCH, and TEST are similar to LOOP in that they represent a branching to an ENTRY Primitive. EXAMPLE may be altered in the following four ways.

3.5.1 PROB. The PROB Primitive is used to indicate that the re-execution of the ACTION "Delay" has only a certain degree of probability.

Since AISIM has no command for directly replacing one Primitive with another, the existing Primitive LOOP must first be deleted.

Enter the command

    DEL 4

where, as before, "4" indicates the location of the Primitive to be deleted. Figure 29 shows the display produced when the LOOP Primitive is deleted. "DEL" is an abbreviation for "DELETE".

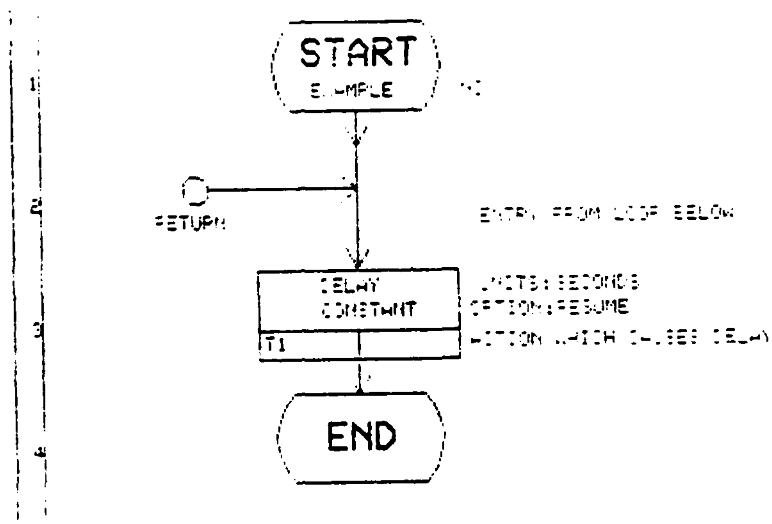


Figure 29. EXAMPLE after Deletion of LOOP Primitive

The PLACE command is used to insert a PROB Primitive between the ACTION "Delay" and the END symbol. Type

    P PROB

The screen will offer the form shown in figure 30.

PARAMETERS FOR PROBABILISTIC BRANCH:

BRANCH TO LABEL:

PROBABILITY OF BRANCH: %

COMMENT:

Figure 30. Form for PROB Primitive

Complete the first field with "Return". The second field should be filled in with the probability of branching, given as a percentage. Suppose there is a 25% chance of branching. Type the appropriate comment, "25% chance of branching". The resulting display diagram is given in figure 31.

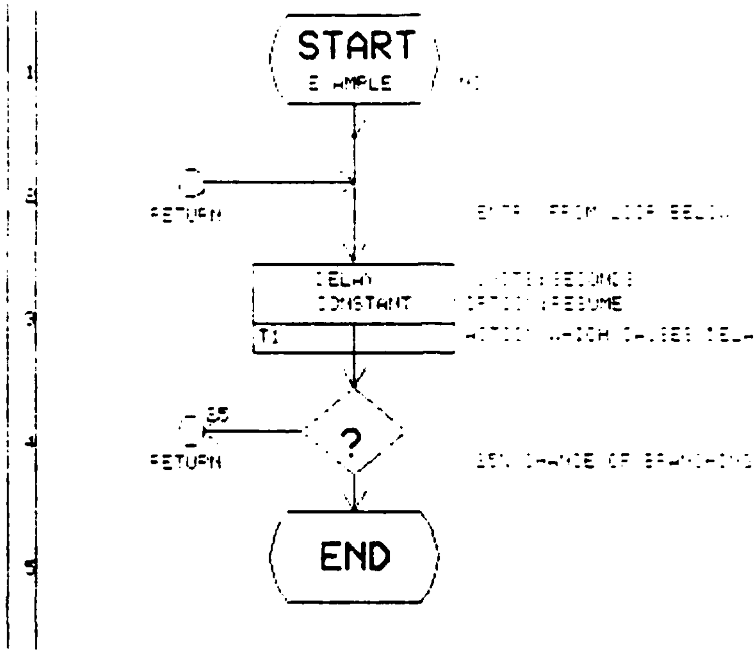


Figure 31. Process with Probabilistic Branch

3.5.2 <u>TEST</u>. Another kind of branching Primitive is TEST. As mentioned earlier, Processes often make use of Resources for which there is competition. The TEST Primitive represents the procedure of ascertaining the availability of a given Resource and branching if that Resource is not available, or continuing if it is. This Primitive does not allocate the Resource. To place the TEST Primitive (after having deleted the PROB Primitive from the latest version of EXAMPLE), type

P TEST

The screen will display the form shown in figure 32.

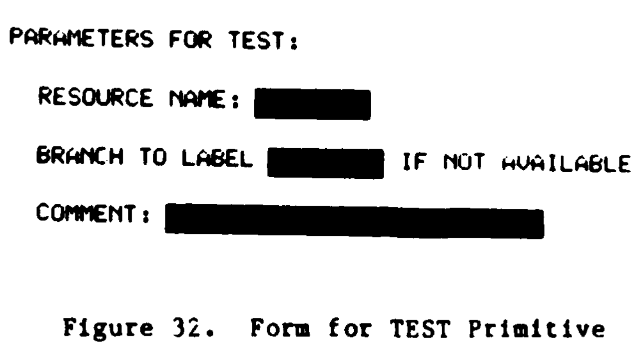


Figure 32. Form for TEST Primitive

In the RESOURCE NAME field, type the name of the Resource whose status is to be ascertained. The LABEL is the ENTRY Primitive to branch to, COMMENT is self-explanatory. If the PROB Primitive in the previous version of EXAMPLE is replaced by TEST (as in this example), EXAMPLE will now appear as in figure 33.

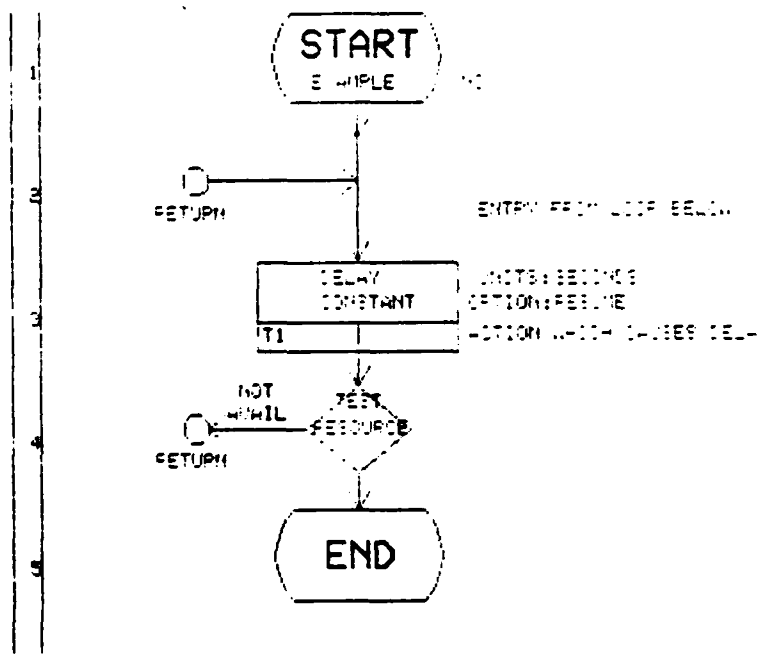


Figure 33. Process with TEST Branching

3.5.3 COMPARE. In addition to probabilistic branching, AISIM also allows for conditional branching less specialized than the TEST Primitive. Most of these branchings will require the COMPARE Primitive. The COMPARE Primitive compares two numerical or alphanumerical values with respect to some relation and branches to a named ENTRY Primitive if the relation holds.

To place the COMPARE Primitive, delete the previously defined TEST Primitive and type

P COMPARE

The screen will display the form depicted in figure 34.

PARAMETERS FOR COMPARE

IF OPERAND 1: QUALIFIER 1:

RELATION:

OPERAND 2: QUALIFIER 2:

BRANCH TO:

COMMENT:

Figure 34. Form for COMPARE Primitive

The fields OPERAND 1 and OPERAND 2 hold the parameters whose values are to be compared. The values may be represented by arbitrarily chosen names of variables (such as Var1 and Var2). They are compared with respect to the following six arithmetical relations indicated by the two letter code:

EQ for "equal to"

NE for "not equal to"

GT for "greater than"

LT for "less than"

GE for "greater than or equal to"

LE for "less than or equal to"

The BRANCH TO and COMMENT labels are now self-explanatory. The two QUALIFIER fields serve several purposes, the most important of which is to allow the comparison of attributes of entities as opposed to simple variables or numerics. The user should for the present disregard the complication posed by these fields and leave them empty. Fill in the OPERAND fields with arbitrarily chosen names of variables, "Var1" and "Var2". If the TEST Primitive is replaced by the COMPARE Primitive with the foregoing information entered on its form, the new version of EXAMPLE will be as displayed in figure 35.

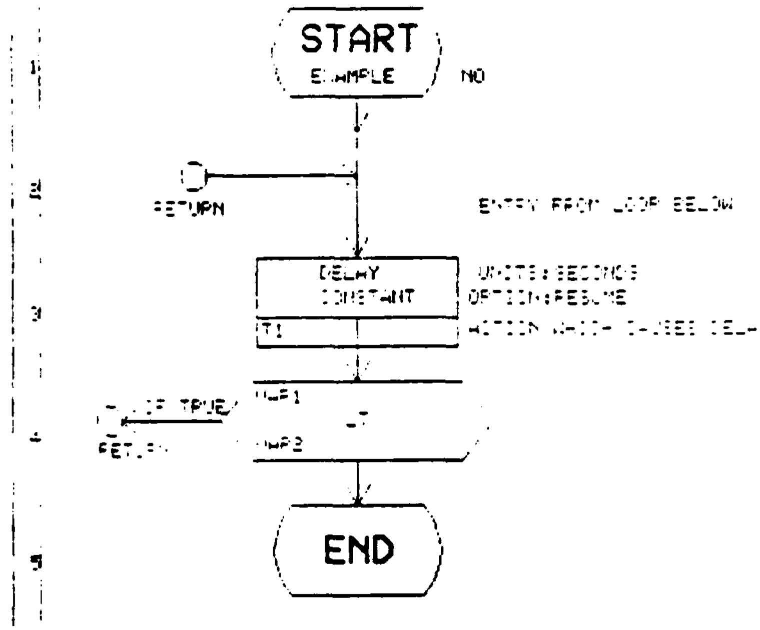


Figure 35. Process with COMPARE Primitive

And thus EXAMPLE is set to return control to the ENTRY Primitive if the value assigned to the variable Var1 is less than the value assigned to the variable Var2. These assignments are presumed to be made elsewhere.

3.6 VARIABLE MANIPULATION

In the previous example of the COMPARE Primitive, note that if the condition solicited is true, i.e., if VAR1 was less than Var2, EXAMPLE would perform the ACTION "Delay" indefinitely. On each occasion in which the comparison is made, the relation will hold and hence the Process will always be instructed to branch to Return. If neither variable changes its value, the Process will continue until it is halted by other causes (such as having a Resource necessary to it allocated elsewhere).

Using two new Primitives, ASSIGN and EVAL, we can alter EXAMPLE so that the ACTION "Delay" does not go on forever but only for a certain maximum time ("Maxtime"). This is accomplished with the ASSIGN Primitive which introduces a new variable for the accumulation of time consumed by the ACTION's execution times and by the EVAL Primitive, which recalculates the value of this accumulated time each time the ACTION is performed.

First, to command AISIM to place an ASSIGN Primitive between the START and the ENTRY, type

    P ASSIGN,2

The screen will now show the form displayed in figure 36.

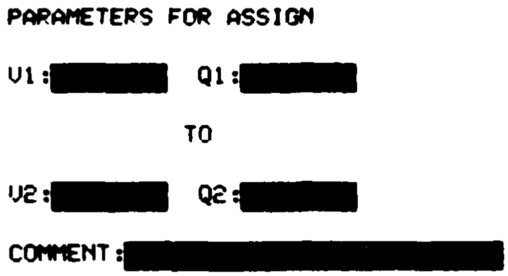


Figure 36. Form for ASSIGN Primitive

For this example disregard the fields labeled Q1 and Q2; they serve the same purpose as do the QUALIFIER fields in the COMPARE Primitive. The purpose of this exercise is to create a temporarily useful, local variable, which we shall call "Acctime" whose value represents the amount of time that has been consumed in the repeated execution of "Delay". At the beginning of the Process the initial value of the variable will be zero. Hence, complete the V1 field with "0" and the V2 field with "Acctime". When this information is entered, the screen will display the graphic representation shown in figure 37.

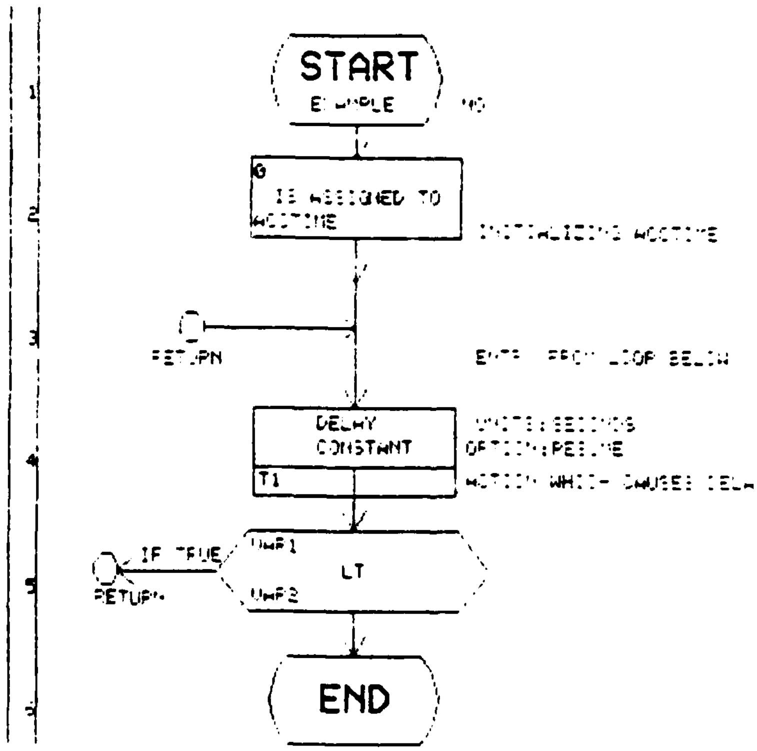


Figure 37. Process with Primitives ASSIGN, ACTION, and COMPARE

To provide an apparatus for updating the variable "Acctime" on each occasion of the ACTION's execution, an EVAL Primitive must be placed between the ACTION and COMPARE Primitives. To do this, type

```
P EVAL,5
```

The screen will display the form shown in figure 38.

```
PARAMETERS FOR EVAL

 SET VARIABLE:

 ARITHMETIC EXPRESSION:



 COMMENT:
```

Figure 38. Form for EVAL Primitive

The SET VARIABLE field holds the name of the variable whose value is to be calculated. The ARITHMETIC EXPRESSION field contains up to four lines of a free-form expression whose result is assigned to the variable in the SET VARIABLE field (see AISIM User's Manual, section 3.9.12). Type an appropriate comment, such as "Evaluating Acctime". The graphic representation of EXAMPLE will be as shown in figure 39.

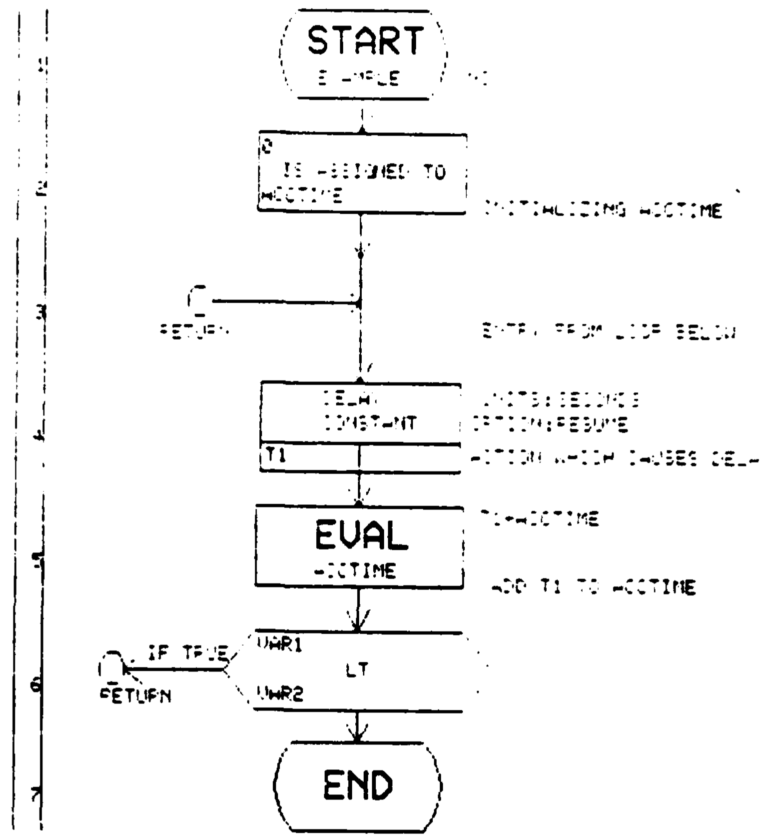


Figure 39. Process with ASSIGN, ACTION, COMPARE, and EVAL

This Primitive adds the current value of the variable "T1" to the value of "Acctime", producing an updated figure for the total time consumed by "Delay".

The Process still requires alteration. The variables presently in the COMPARE Primitive must be changed from Var1 and Var2, respectively, to "Acctime" and "Maxtime". To do this, we must edit the COMPARE Primitive by typing

C 6

This command tells AISIM that you wish to alter one or more of the previously defined parameters in the Primitive at location 6. The screen will display the form for the Primitive. It can be altered simply by writing over the existing information. When this is done and the form is "entered", EXAMPLE will satisfy the specifications for its alteration. Its graphic representation will be as in figure 40.

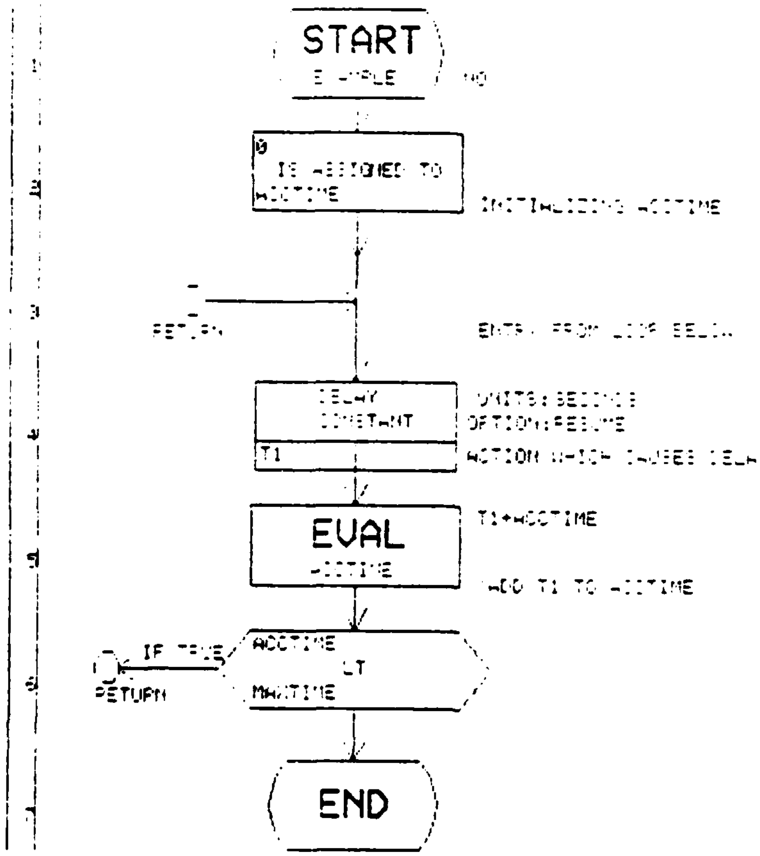


Figure 40. Process with Comparative Branching

3.7 ITEM MANIPULATION

Another group of Primitives is categorized under the headings Queue Handling and Item Handling. These include CREATE, DESTROY, FILE, FIND and REMOVE. The Primitives CREATE and FILE will be used in this example. (Consult the AISIM User's Manual for information on the Primitives DESTROY, FIND and REMOVE.)

Consider the first version of EXAMPLE which consisted of the single ACTION Primitive "Delay". Suppose now that we conceive of EXAMPLE as a Process which gives rise to new data elements--messages, information, potential communications. This function of the Process may be represented by means of the CREATE Primitive, which represents the introduction of Items--the AISIM modeling entity that represents transient data elements--into the modeled system. To place the Primitive CREATE below the ACTION Delay in EXAMPLE type

    P CREATE

The form for CREATE is shown in figure 41.

PARAMETERS FOR CREATE

ITEMS TO BE CREATED ARE:

COMMENT:

Figure 41. Form for CREATE Primitive

Complete this form with the names of the Items to be created. Enter the Item name "Msg" and an appropriate comment, "Transient Data Element". EXAMPLE will now appear as indicated in figure 42.

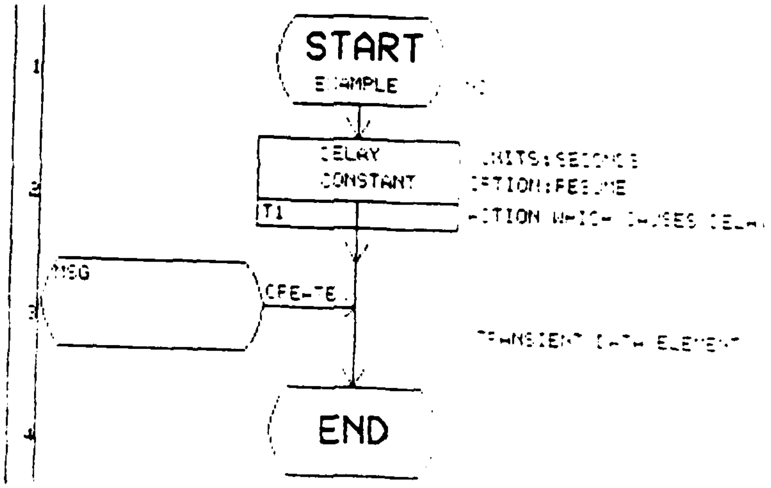


Figure 42. Item Creating Process

Items—transient data elements—can also be filed in holding areas called Queues with the FILE Primitive. To place a FILE Primitive below the CREATE Primitive in EXAMPLE, type

P FILE

The form for FILE is as shown in figure 43.

PARAMETERS FOR FILE:

FILE ITEM NAME: OPTION: LAST ON QUEUE:

COMMENT:

Figure 43. Form for FILE Primitive

Complete the field FILE ITEM NAME with "Msg". The OPTION field tells where in the Queue the Item is to be placed. This location is specified relative either to absolute locations on the Queue ("FIRST" and "LAST") or relative to some other Item already on the Queue ("BEFORE" and "NEXT")*. The OPTION field will have as a default parameter LAST. In the ON QUEUE field enter "Msg_que". The graphic representation of this Process is indicated in figure 44.

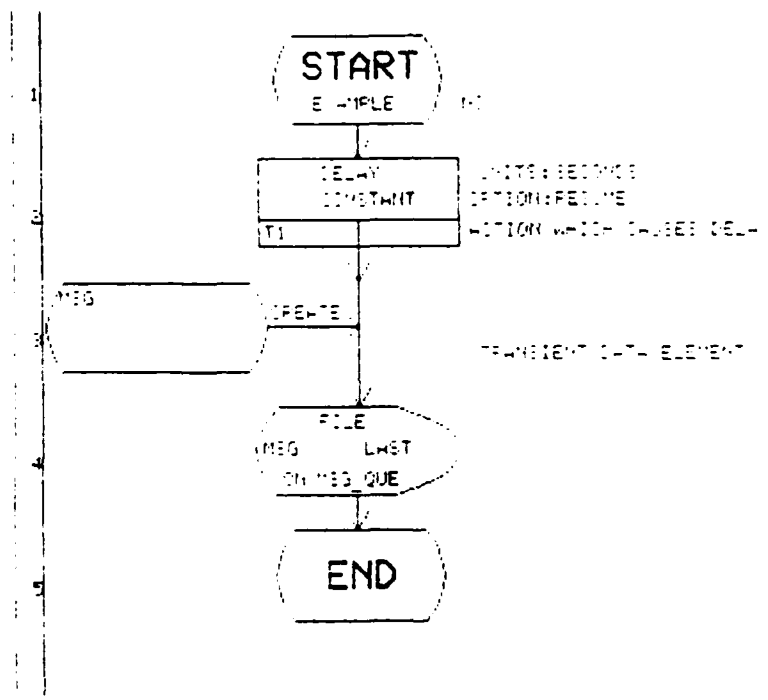


Figure 44. Process which Creates and Files Message Items

*The method by which the system identifies the Item relative to which other Item is to be placed on a Queue (with the OPTION BEFORE or NEXT) need not concern us here. For more on this, see AISIM User's Manual, section 3.9.13.

3.8 RELATIONS AMONG PROCESSES

This section deals with the relationships that the execution of Processes bear to one another in an AISIM Model, and how one Process, and its execution, affects the execution of another. Processes affect one another's execution in four ways:

1. By sending Items that trigger the execution of another Process.

2. By triggering the execution of another Process through a CALL Primitive where the CALL may or may not pass parameters.

3. By competing for and obtaining use of a Resource needed by another Process.

4. By resuming (with the RESUME Primitive) a Process which at some time suspended itself.

To understand how parameter and Item "passing" affect the execution of a Process, consider the form completed in the first version of EXAMPLE. In the form presented as a result of the command to edit a Process (i.e., E PROCESS,EXAMPLE,NEW), in the START field TYPE, the choices included "STD", "PARM" and "ITEM", standing for, respectively, "standard", "parameter-passing" and "Item-passing". These options are distinguished from one another in the following way. A Process can, before it is fully designed, be thought of as a "black box" whose internal workings are unknown. If the Process is conceived to be one that performs its function without having to be given anything in the way of information or data elements it will be a standard Process. If the Process requires certain data elements—discrete, countable entities--in order to execute, then it is an "Item-passing" Process. Finally, if the Process uses values of variables local to another Process, it is a parameter-passing Process.

For the first example, consider Item Passing Processes. In this exercise, delete the FILE and CREATE Primitives from EXAMPLE. To change EXAMPLE from a standard Process, as it now is, to an Item-passing Process, type

    C 1

The screen will display the form originally filled out for EXAMPLE. Type "ITEM" over the existing "STD" in START field TYPE. Entering this, the screen will now show this secondary form on which Items needed by this Process are to be typed, as shown in figure 45.

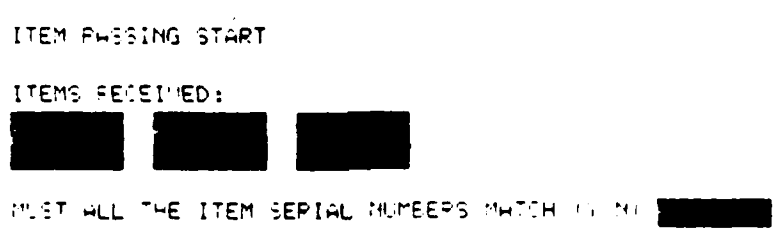


Figure 45. Secondary Form for Process

Type the single Item name "Msg" in the upper left field and type "N" in the match field. Entering this changes the Process representation so that it appears as in figure 46.

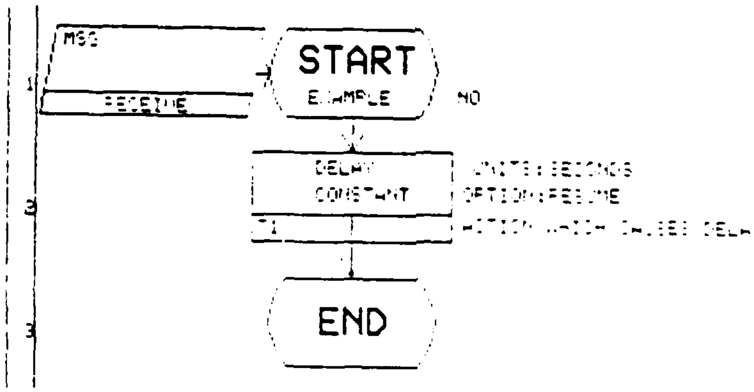


Figure 46. Item Passing Process

The figure to the left of the START figure indicates that the Process starts when, and only when, the Item MSG is delivered to it from some other Process.

None of the Primitives in the categories Item Handling and Queue Manipulation represents the delivery of Items to a Process. This delivery function is accomplished by the SEND. To exemplify SEND, a new Process, called "EXAMP_2", must be created. EXAMP_2 triggers the execution of EXAMPLE by delivering Items to it. For this example consider a Process identical except in name to the original EXAMPLE with the single ACTION Delay as depicted in figure 47.

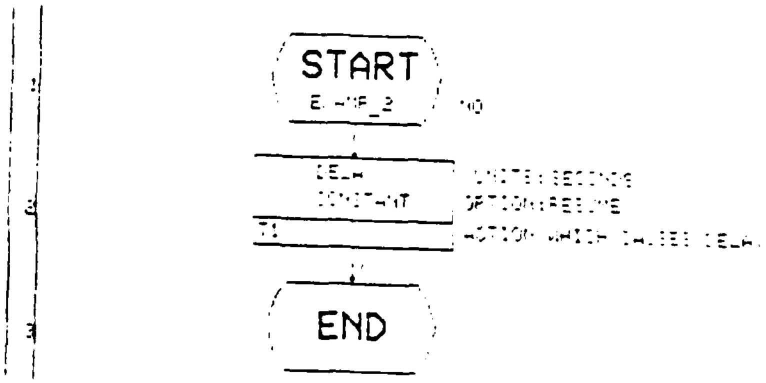


Figure 47. Process with ACTION Primitive

Type the command

    P SEND

The screen will display the form shown in figure 48.

PARAMETERS FOR SEND

SEND ITEMS TO

ITEMS TO BE SENT ARE:

COMMENT:

Figure 48. Form for SEND Primitive

Complete the SEND ITEMS TO field with "Example". In the first field of ITEMS TO BE SENT, type "Msg". Enter the comment "Sending Messge Item". Figure 49 shows the graphic representation of the Process that will appear on the screen.

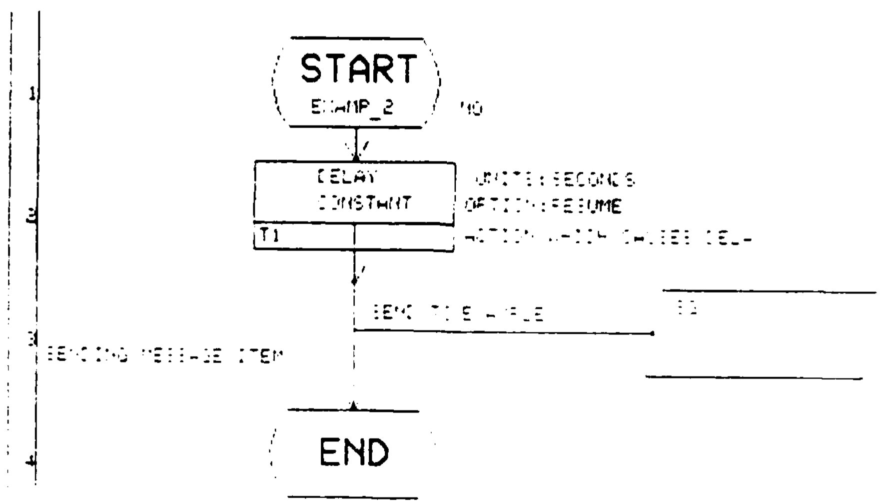


Figure 49. Graphic Representation of EXAMP-2

EXAMP_2 now triggers EXAMPLE by delivering to it Items required for its execution. The Item is automatically created by the SEND Primitive. An Item-passing Process may only be initiated through the SEND Primitive in some other Process, although the Items needed, and hence the Items sent, may be distributed among several Processes or several stages of a single Process.

3.9 RESOURCE ALLOCATION

As mentioned earlier, Processes in an AISIM model frequently make use of Resources. A Resource has a finite capacity which will limit the number of Processes it can accommodate at the same time. The five Primitives which relate to the allocation of such Resources are ALLOC, DEALLOC, RESET, LOCK and UNLOCK.

ALLOC and DEALLOC signal the allocation and deallocation of a Resource by the Process in which they appear. To place the ALLOC Primitive above the ACTION Primitive in EXAMPLE, type

P ALLOC,2

To place a DEALLOC Primitive just above the SEND symbol in EXAMP-2, type

P DEALLOC,4

The forms for these two Primitives are shown below in figure 50.

PARAMETERS FOR ALLOCATE:

ALLOCATE RESOURCE NAME:

NUMBER OF UNITS REQUESTED: 1

PARTIAL/ALL ALLOCATION: PARTIAL

ALLOCATION PRIORITY: $PRIORT

COMMENT:

PARAMETERS FOR DEALLOCATE:

DEALLOCATE RESOURCE NAME:

NUMBER OF UNITS DEALLOCATED: 1

COMMENT:

Figure 50. Forms for Primitives ALLOC and DEALLOC

In each case, enter the name of the Resource to be allocated or deallocated, such as "Cpu", in the field provided. The field NUMBER OF UNITS REQUESTED holds the number of units of this Resource to be allocated or released. The PARTIAL/ALL field specifies the type of allocation scheme. Partial allocation will allocate Resources as they become available. All allocation means all of the units must be available at once. The ALLOCATION PRIORITY field specifies the priority at which Resources are allocated. Enter "1" for number of units and "$Priorty" for priority. "$Priorty" means to use the priority the Process was invoked with. Enter the appropriate comment, "Obtaining Cpu" or "Releasing Cpu" in the COMMENT field.

Placing these Primitives in EXAMP_2 (one above the ACTION and one below), produces a graphic representation like that shown in figure 51.

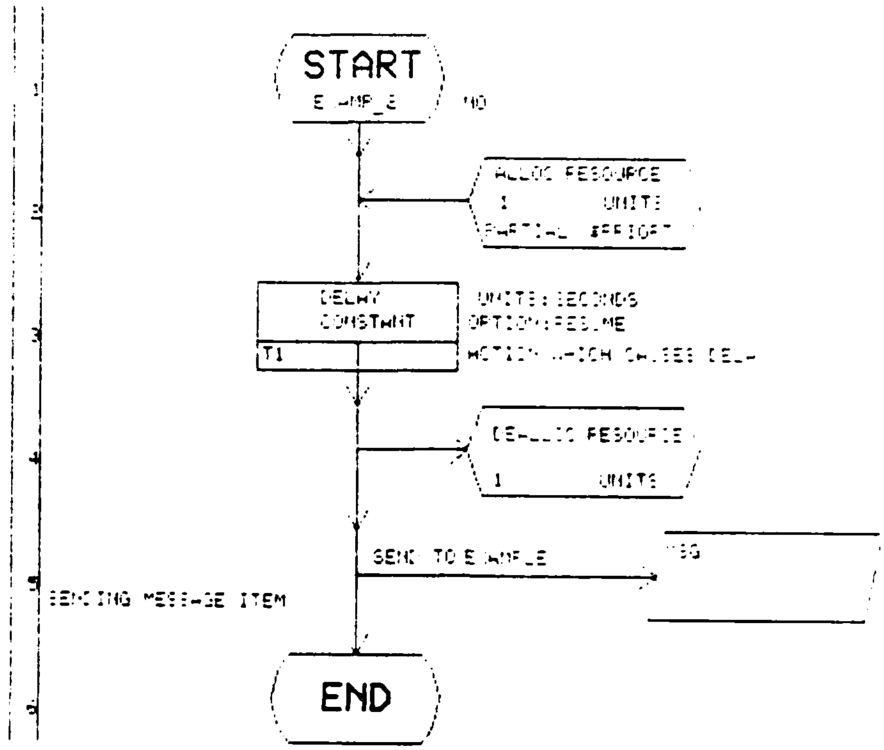


Figure 51. Process which Allocates and Deallocates a Resource

Allocating a Resource does not normally insure the uninterrupted availability of that Resource to a Process. Any Resources may be usurped by an ALLOC request with a higher priority. If the Process being modeled is one which, once begun, cannot be interrupted, the Primitives LOCK and UNLOCK must be used.

To obtain the forms for these Primitives,

P LOCK,n

or

P UNLOCK,n

where n is the position in the Process where the Primitive is to be placed. The forms for these Primitives are shown in figure 52.

PARAMETERS FOR LOCK:

COMMENT:

PARAMETERS FOR UNLOCK:

COMMENT:

Figure 52. Forms for Primitives LOCK and UNLOCK

These Primitives, if placed above the ALLOC Primitive and below the DEALLOC Primitive in EXAMP_2 would give a graphic representation like that shown in figure 53.

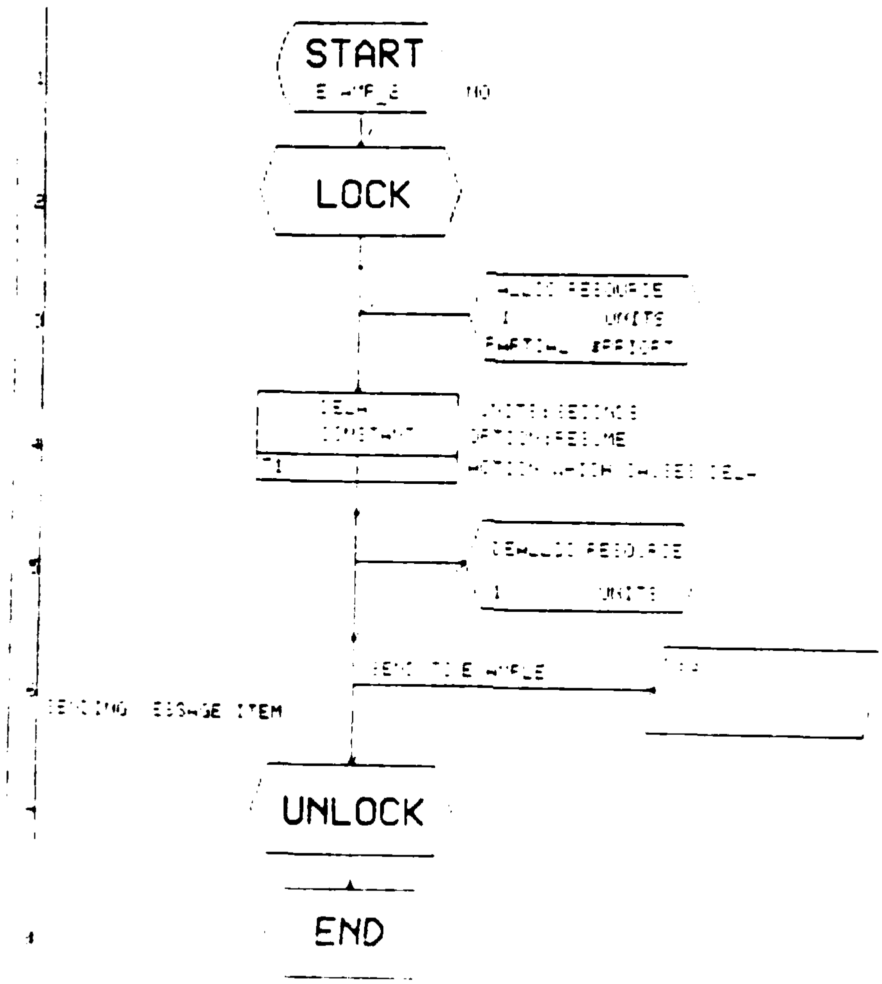


Figure 53. Process with Protected Resources

One final way to affect a Resource is through the RESET Primitive. It is used to reset the capacity of a Resource, where "capacity" is a measure of the number of Processes it will accommodate (support) at one time. For details on its use, see the AISIM User's Manual, section 3.9.20.

3.10 CALL

The function of the CALL Primitive is similar to that of the SEND Primitive, but whereas the SEND Primitive triggers Item-passing Processes, the CALL Primitive triggers both standard Processes and parameter-passing Processes. Thus, to understand how CALL works requires a brief discussion of parameter-passing Processes.

A parameter-passing Process is one that is "given" values for input variables and "returns" values for output variables. To create a paramater-passing Process, one would type "PARM" in the field START TYPE in the original form for Process. Entering this information on the Process form yields the secondary form shown in figure 54.

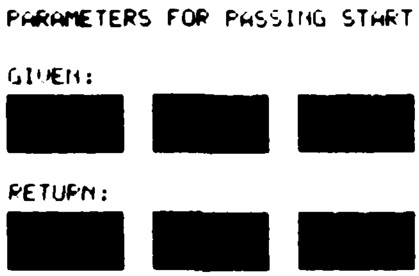


Figure 54. Secondary Form for Parameter-Passing Process

On the form in figure 54, the user types the variables whose values are passed to the Process and the variables whose values are passed back.

The CALL Primitive values, i.e., parameters, are passed (given) to a called Process and returned to the calling Process. Parameter passing can occur only through the use of a CALL Primitive. A CALL Primitive is placed in a Process by typing

P CALL

The form for CALL is shown in figure 55.

PARAMETERS FOR CALL

CALLED-PROCESS NAME:

WAIT/NOWAIT/BLOCK: NOWAIT PRIORITY:

GIVEN:

RETURNS:

COMMENT:

Figure 55. Form for CALL Primitive

The field CALLED-PROCESS NAME asks for the name of the Process to be triggered. The field PRIORITY determines the priority associated with the called Process which may be used in cases of Resource contention. (It can be overridden by the priority specified in an ALLOC Primitive.) The GIVEN and RETURNS fields hold the local variables whose values are passed to and from the called Process. A CALL Primitive may trigger a standard Process and hence these fields may be empty. The COMMENT field is self-explanatory. The field labeled WAIT/NOWAIT/BLOCK determines whether the calling Process will wait for the called Process before continuing execution or will continue to execute independently of it. The reader is referred to the AISIM User's Manual for details on their use.

4. REMAINING MODEL ELEMENTS

Although the Processes and the Architecture are core modeling elements, their specification does not complete the task of model construction. They must be supplemented by definitions of other elements. These elements are grouped into two categories. The first category consists of those entities explicitly referred to in Processes, namely, Actions, Constants, (global) Variables, Tables, Queues and Resources. The second category consists of the two entities that are used to represent the impact of the environment on the modeled system. All these remaining entities are defined at the DUI level of AISIM operation.

The following two sections briefly describe the parameters, significance and principle commands associated with these remaining entities.

4.1 ACTIONS

Any ACTION Primitive placed in a Process must have a corresponding Action entity defined outside the Process. AISIM automatically creates an Action entity whenever the user places an ACTION Primitive in a Process. However, if the user wishes to change the description for the Action entity, this can be accomplished by typing

    E ACTION,ACTION NAME

The form for the Action entity is shown in figure 56.

```
ACTION:
    DESCRIPTION:
```

Figure 56. Form For Action Entity

The ACTION field should hold the name of an Action referenced in some ACTION Primitive. The field DESCRIPTION is for any convenient reminder of what the Action represents. Normally a user will not need to modify Action entities, and their maintenance can be left to the AISIM system.

4.2 RESOURCES

As mentioned earlier, any Resource mentioned in a Process--through the ALLOC, DEALLOC, or RESET Primitives--must be defined separately in the DUI. To create a new Resource, type

    E RESOURCE,NAME,NEW

The screen will display the form shown in figure 57.

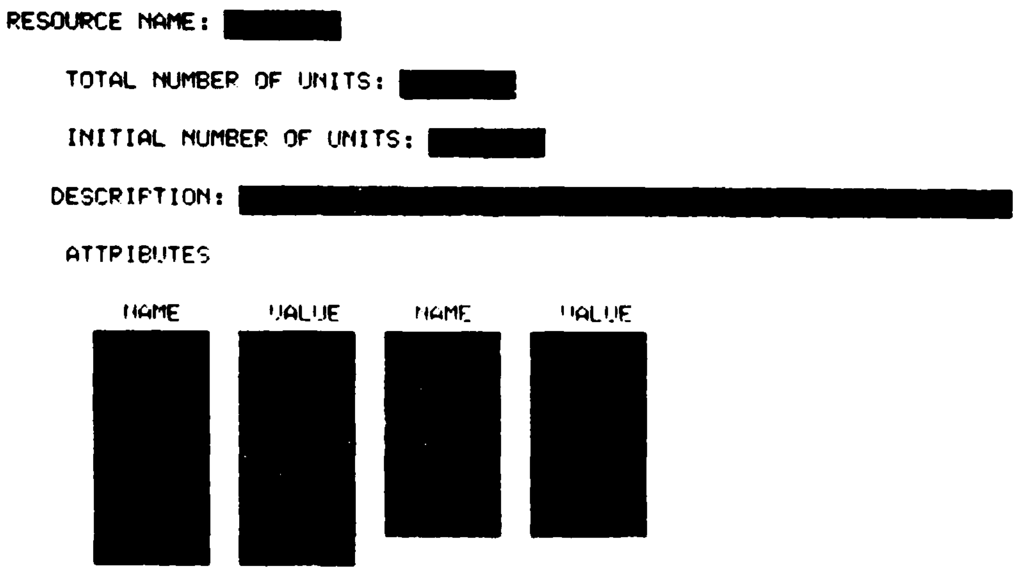


Figure 57. Form For Resource Entity

Complete the first field, RESOURCE NAME, with the name by which it is referred in any Process. The fields TOTAL NUMBER OF UNITS and INITIAL NUMBER OF UNITS indicate, respectively, the maximum number of Processes the Resource can accommodate at any one time and the number of Processes it can accommodate at the beginning of a simulation run (i.e., before being increased or decreased by the RESET Primitive). Enter the appropriate numbers. DESCRIPTION has its usual function. Type an appropriate description in the field provided. The fields under ATTRIBUTES indicate whether the Resource has any attributes associated with it.

Up to fifteen attribute names may be entered with their initial values.

Though all Resources referred to require separate definitions, some Resources are defined automatically. For each node or link created in a model's network architecture, a Resource definition of the same name with default parameters is automatically written into the database. In other words, all nodes and links are identified with Resources. Thus, after an architecture has been created, the commands

```
E RESOURCE,nodename
        or
E RESOURCE,linkname
```

can be issued without having to indicate that the Resource entity is new (with "NEW"). Typically, however, not all of a system's Resources will be represented in the architecture and not all of the Resources automatically created in ADE will have any positive role in the operation of the model. That is, such automatically defined Resources need not be invoked in the Process Primitives. Importantly, if an operative Resource is to be identified with an architectural element, it should be defined first in ADE and thereafter edited to provide it with suitable parameters (on the assumption that the default parameters are incorrect). ADE will not allow the definition of a node or link whose name is identical with that of a Resource already in existence.

4.3 QUEUES

Not all the Queues functioning in a system model need be defined by the user, since many are implicit in the operation of the system. The general rule is that any Queue manipulated by the FILE, FIND or REMOVE Primitives must be given a separate definition in the DUI, with the exception of a cross-reference set.

The cross-reference sets are explained in the AISIM User's Manual, section 3.5.2.

To define a new Queue, type

    E QUEUE,NAME,NEW

The form for this entity is shown in figure 58.

```
QUEUE:            SIZE: INFINITE

DESCR:
```

Figure 58. Form for Queue Entity

The three fields should be filled in with, respectively, (a) the name of the Queue as found in the FILE, FIND, or REMOVE Primitive which uses the Queue, (b) the maximum number of Items that can be placed in it (the default value for which is "infinite") and (c) any useful reminder of the Queue's role in the modeled system.

4.4 CONSTANTS AND VARIABLES

Constants differ from global Variables only in that they do not change their values during the simulation exercise of a model. Once a value has been assigned to a Constant and a simulation is begun, its value is unchanging. Accidental attempts to alter the value of a Constant through the EVAL or ASSIGN Primitives will yield an execution error message. The forms for Constants and Variables are quite similar and are called up by issuing the command EDIT CONSTANT or EDIT VARIABLE and then giving the name of the Constant or Variable.

The forms for Constants and Variables are shown in figure 59.

CONSTANT:

VALUE:

DESCRIPTION:

VARIABLE:

VALUE:

DESCRIPTION:

Figure 59. Forms for Constant And Variable

The fields CONSTANT and VARIABLE call for the entities' names. The VALUE fields call for numerical values for Constants and alpha-numerical values for Variables, and the DESCRIPTION fields call for any description.

4.5 LOADS AND SCENARIOS

The effect of the environment on a model is represented collectively by Loads and Scenarios. The relationship between Loads and Scenarios is this: Loads specify a number of Process triggerings to take place sometime during the simulation exercise of a model. Loads do not specify when the Process triggerings are to take place. They specify the distribution of the time passing between triggerings of the Process. Scenarios specify a collection of Loads and/or individual Processes together with a schedule indicating when the specified Loads or Processes are to be initiated.

To define a Load, type

```
E LOAD,NAME,NEW
```

The form for the Load Entity is shown in figure 60.

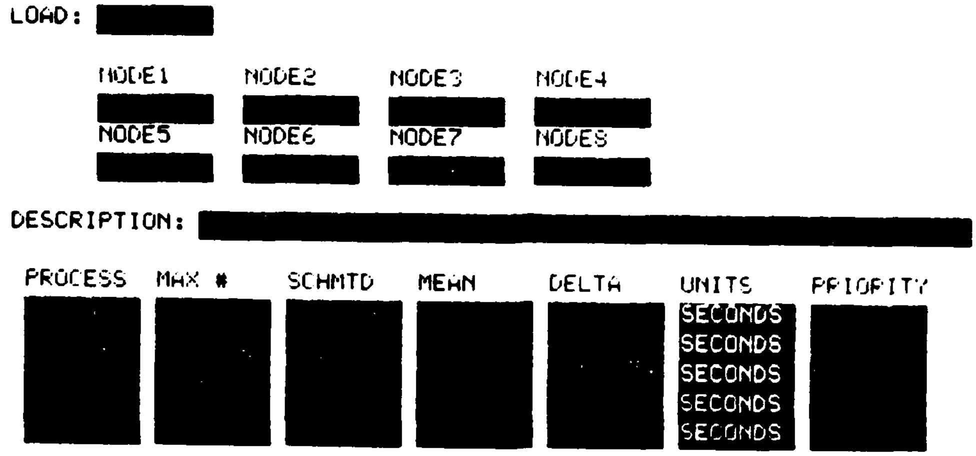


Figure 60. Form for Load Entity

The LOAD field holds the name of the Load. The fields labeled NODE1 through NODE8 indicate the architectural nodes in which the Processes named in the Load take place. The field DESCRIPTION is for any helpful description.

The field labeled PROCESS holds up to five names of Processes. The fields SCHMTD, MEAN, UNITS and DELTA together define the statistical method of distribution to be used in scheduling the Process triggerings. SCHMTD holds the name of the distribution method, MEAN holds the average time between Process initiations (in terms of the UNITS specified) and DELTA is a second numerical parameter used to specify variation about the mean, if applicable. The field MAX # indicates the maximum number of Process instances to be initiated by this Load.

The Scenario entity defines the impact of the environment on the system for the entire simulation exercise of a model. In it the user specifies a number of "periods" into which a simulation exercise is to be divided, together with a uniform length each period is to have. The user then specifies a collection of Loads or Processes to be initiated at a specified time during the simulation. A priority is also given for each Process.

To define a Scenario, type

```
E SCENARIO,NAME,NEW
```

The form for the Scenario entity is shown in figure 61.

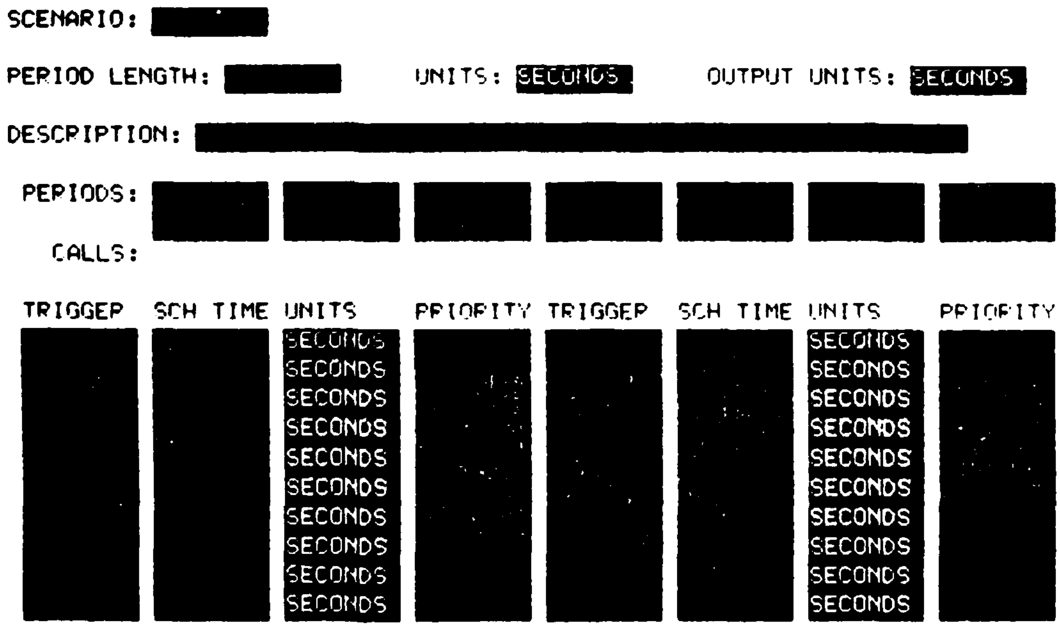


Figure 61. Form for Scenario Entity

The field SCENARIO holds the name of the entity. PERIOD LENGTH is the length of each period. The field UNITS is the time unit in which the period length is specified. The OUTPUT UNITS is the time unit to use when generating the simulation output reports. The 14 fields labeled PERIODS are used to indicate the number of periods the Scenario is to have. The number of periods in the Scenario is determined by the number of these fields in which an entry is made. Any characters may be typed in these fields.

The fields labeled TRIGGER take the name of the Load or Process to be initiated. The field SCH TIME indicates the time at which the Load or Process named immediately to the left is to be initiated. The field UNITS is the time unit for scheduling. The field PRIORITY is used to assign a priority to the named Process; it is ignored for a Load and should be left blank.

5. A WORKING EXAMPLE

This section documents the construction of an AISIM model that can be run through simulation tests and analyzed in the subsequent chapter. The model will be a representation of the transmitter/receiver relationship, an element of any communication system.

The transmitter/receiver relation modeled here is of the "polling" or "mailbox" type, as opposed to the "interrupt" type. In it, one transmitting Process generates messages and delivers them to a buffer. There the messages await treatment from a receiving Process. The transmitting and receiving Processes are not in direct communication with one another. Rather, the transmitter broadcasts messages according to need, and the receiving Process reads them from the buffer at intervals in accordance with expected need. In the system envisioned, transmission is randomized in two respects, (1) in the lengths of transmitted messages and (2) in the intervals between transmission. Reception is undertaken at regular intervals and the time consumed in processing a message is a linear function of its length.

The origination of a message in the transmitting Process will be represented by the creation of an Item (through the CREATE Primitive). The Item will have a variable attribute which will represent its length. Since the length will be randomized over a range of approximately 700 bytes, some mechanism must be incorporated for altering the variable attribute of each data Item (i.e., message). This is accomplished by (1) generating a random number in the range [0,1] subsequent to the creation of each Item, (2) multiplying the random number by twice the average message length and (3) assigning the number so obtained to the message length. This figure will then be used to calculate the time taken to send the message to the buffer (where it will be available to the receiving Process). Through an ACTION Primitive, the clock is then updated by the amount calculated.

In this system the buffer will not be manipulated by both the receiving and the transmitting Processes at the same time, so the buffer is considered a Resource and its allocation and deallocation by the ALLOC and DEALLOC Primitives will prevent it from being accessed simultaneously by both Processes.

5.1 DEFINING PROCESSES

This description of the transmitting Process gives the steps of its execution. The transmitting Process:

(1) Starts

(2) Allocates one unit of a Resource BUF1 representing the buffer

(3) Creates a message, represented as an Item called "Msg"

(4) Generates a random number between 0 and 1 and then multiplies it by twice the average message length

(5) Assigns the number obtained in the previous step to the Item attribute representing the message length

(6) Calculates the delay time which is proportional to the message length (i.e., an amount equal to the message length multiplied by the transmission rate in seconds per byte)

(7) Delay for the calculated amount of time

(8) Delivers the message Item to the Queue called Buffer through the FILE Primitive

(9) Releases the Resource BUF1 representing the buffer

Figure 62 shows the Process flowchart derived from this description.

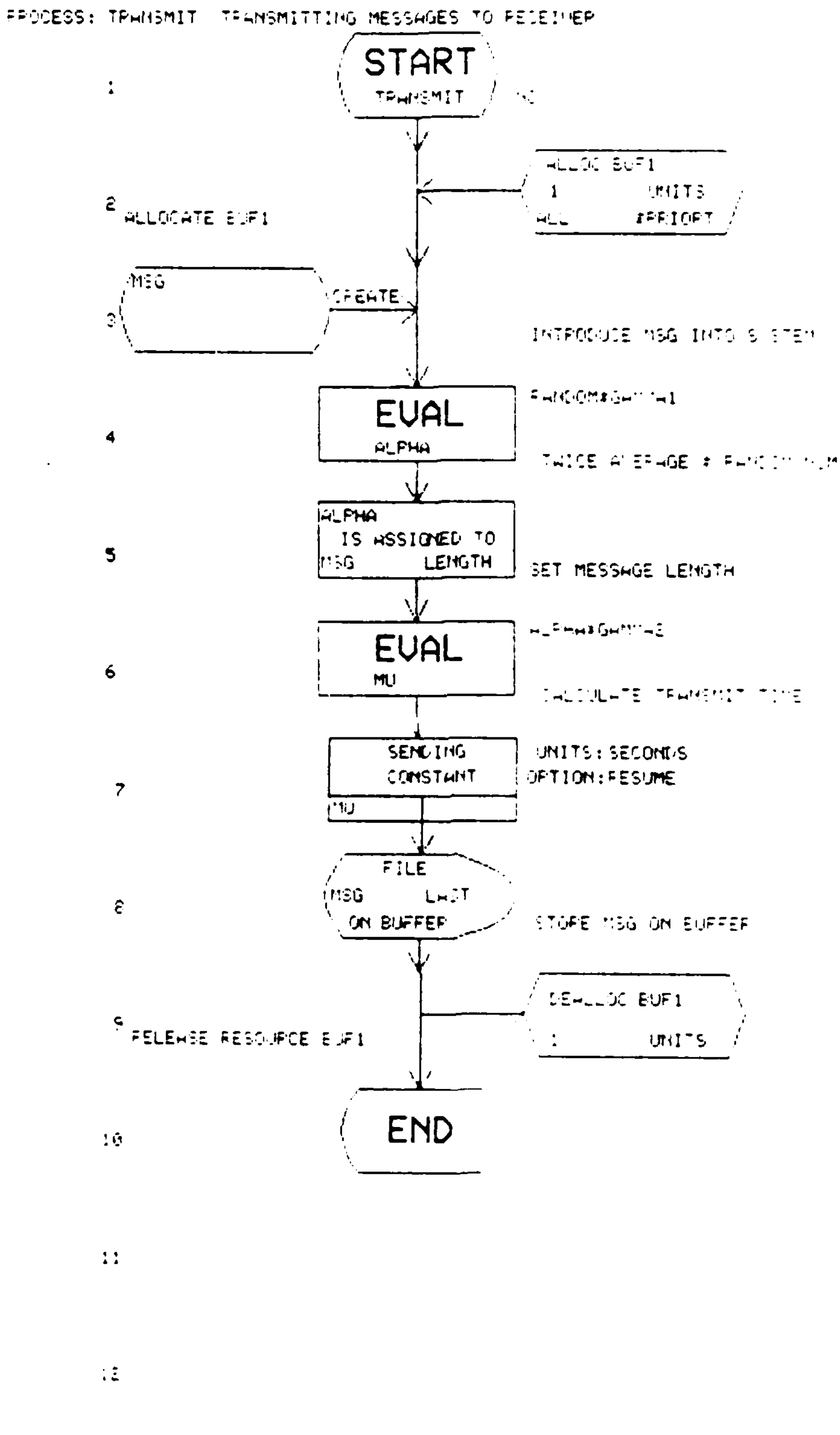


Figure 62. Transmitting Process

The receiving Process will first determine whether or not the buffer is being manipulated by the other Process by testing for utilization of the Resource call BUF1. If the Resource is in use, the Process will abort by branching to the END symbol. If BUF1 is free, the Process will read the next message from the buffer, and calculate a receiving time in roughly the same way that the transmitting time for that same Item was determined in the transmitting Process. The clock is then updated by the amount of time calculated.

This description can be expanded into more specific design requirements. The receiving Process

(1) Starts.

(2) Tests for the availability of the buffer by determining whether or not the Resource is in use through the TEST Primitive. If so, the Processes execution will branch to the END symbol.

(3) The next message Item on the Queue called Buffer is read via the REMOVE Primitive.

(4) If there is nothing on the buffer, Process execution, as in step (2), branches to the END. This step is represented by a COMPARE Primitive.

(5) The message length is assigned to a local variable through the ASSIGN Primitive.

(6) A receiving time is calculated to be proportional to the message length (i.e., equal to the message length times some reception speed in seconds per byte).

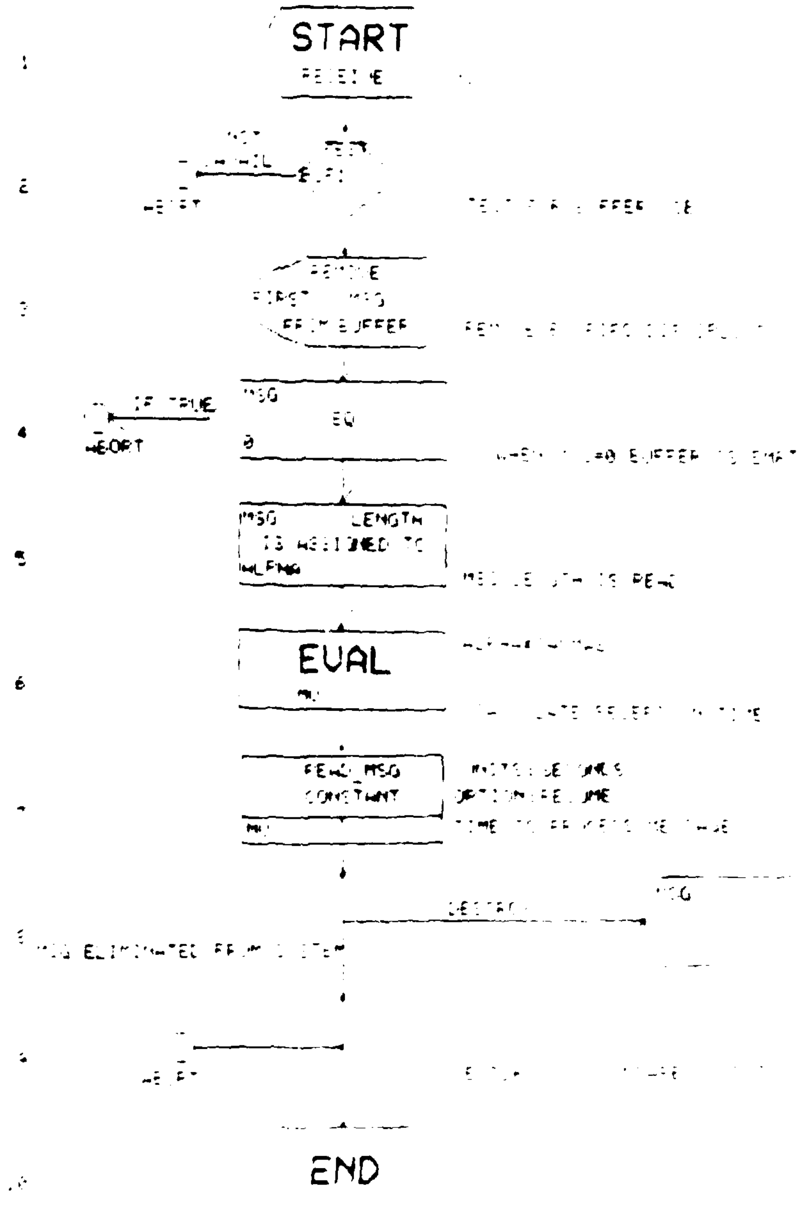
(7) The clock is updated through the ACTION Primitive in the amount required to receive the message.

(8) The message Item, having been read, is eliminated from the system through the DESTROY Primitive.

(9) An ENTRY Primitive is inserted just before the END symbol of the Process to indicate where execution is to resume from the branchings in steps (2) and (4).

Figure 63 shows the flowchart representation of the Process derived from these requirements.

PROCESS: RECEIVE   RECEIVE MESSAGES FROM TRANSMIT



5.2 REMAINING MODEL ELEMENTS

The remaining model entities must now be defined. These include all the entities mentioned in any Process Primitive. These are the following:

1) The Queue named "Buffer" onto which messages are filed.

2) The Resource Bufl which represents a device to protect the buffer against manipulation by two Processes at once.

3) The Item Msg, each instance of which is to represent a message transmitted onto the Queue.

4) The global Variables named Gammal, which is twice the average message length, and Gamma2, which is the transmission rate.

5.2.1 RESOURCE DEFINITIONS

The Resource BUFl is given proper parameters. It will have only one initial unit and will have a maximum of one. It will retain the default of no attributes and a cost of zero. An appropriate description is: "Resource Associated With Buffer".

5.2.2 QUEUE DEFINITIONS

The Queue called "Buffer", which is accessed by the FILE and REMOVE Primitives, will retain its default size of "Infinite". A helpful description is "Buffer On Which Messages Are Stored".

5.2.3 ITEM DEFINITION

The Item Msg which represents messages transmitted and received will have one attribute called "Length". Its initial value will be the literal "$Length", since the value of this attribute will always be assigned within the Process that transmits it to the buffer.

5.2.4 VARIABLE DEFINITION

The Variables "Gammal" and "Gamma2" are defined with initial values [illegible] and .002 respectively. These values are used in calculating the transmition and reception time. Gammal is twice the average message length, and Gamma2 is the transmition rate.

5.3 LOADS AND SCENARIOS

Finally, we must define the hypothetical conditions to which the modeled system will be exposed. Six Loads are defined for this model. [illegible] each trigger the transmitting Process. L11, L22 and [illegible] each trigger the receiving Process in a schedule of expected need associated with [illegible] [illegible]. The triggerings of the transmitting Process are randomized, whereas the triggerings of the receiving Process are scheduled at regular intervals. The complete Load definitions are found on pages [illegible] of the Model Verification Report which appears in Appendix B.

The Scenario for this model consists of six periods. The Loads are distributed throughout the simulation period as follows: each pair of Loads is triggered at intervals of 200 units on the simulation clock. The complete definition is found on page 5 of the Model Verification Report in Appendix B.

6. SIMULATION EXERCISES OF AISIM MODELS

The model is now ready to be run through a simulation exercise to determine its behavior under the defined environmental conditions. To begin this exercise, enter the Analysis User Interface (AUI) from the AISIM READY level by typing

```
A P(projectname)
```

Projectname is the name of the model we wish to expose to a simulation exercise. The user will be prompted with information that will look something like that shown in figure 64.

```
CURRENT PARAMETERS IN EFFECT:
VERSION:       PRODUCTION VERSION 5.0
TERMINAL:      HP
PROJECT:       TEST
USER:          [USER]
ENTER YES TO PROCEED, NO TO ABORT...
```

Figure 64. Information Displayed on Entering The AUI

After declining the abort prompt by typing

```
YES
```

AISIM will ask the user if the current simulation run should be commented with the prompt

```
DO YOU WANT TO ADD A DESCRIPTION FOR THIS RUN?(Y/N)
```

If the user wishes to comment the simulation run, a form will be displayed. The form provides for the entry of 10 lines of information which will be placed at the beginning of the output report.

Following the translation of the model, the user is in a position to issue commands before the execution of the simulation.

6.1 INITIALIZING A MODEL

If more than one Scenario has been defined, the system will ask

WHICH SCENARIO DO YOU WISH TO TRANSLATE?

Type the name of the Scenario that defines the environmental conditions to which the model is to be subjected. For this model, we have defined only one Scenario so the program will perform model initialization. If no errors are detected at this stage the computer will prompt with

NO ERRORS DETECTED DURING MODEL TRANSLATION
YOU MAY NOW ENTER COMMANDS

If an error had been made, AISIM would have prompted with

ERRORS DETECTED IN MODEL TRANSLATION

This prompt indicates that some aspect of the model definition is in error. If such is the case, determine where the errors are, see Appendix B of the AISIM User's Manual for a description of the error messges, and return to the DUI to correct them. The matter of getting to the DUI has already been covered in previous chapters. For this example, assume that the AISIM model is properly defined.

6.2 DEFINING PLOTS

Two choices are available at this point: Proceed to the simulation exercise of the model or request that graphs of some of the activities monitored during the simulation be defined so that they can later be inspected at the terminal.

For example, in the model under consideration, one of our main concerns is to determine whether the buffer onto which the transmitting Process places messages (and from which the receiving Process retrieves the messages) reaches some maximum burden or whether it shows a tendency to infinite queueing. To produce a graph of the behavior of the buffer we type

    DEF QUEUE,BUFFER

The screen will display a selection of the aspects of the behavior of a Queue for which a graph can be defined. These are shown in figure 65.

```
ATTRIBUTES (PLACE AN X NEXT TO ONLY ONE)

    NUMBER IN QUEUE
    NUMBER BLOCKED
    TIME IN QUEUE
    TIME BLOCKED
```

Figure 65. Aspects of Queue Behavior

To define a graph showing the number of Items in the Buffer, we would enter an "X" for "NUMBER IN QUEUE". The screen would then display the options for defining the type statistic on the number of Items in the queue. These options are shown in figure 66.

```
STATISTICS (PLACE AN X NEXT TO ONLY ONE)

      CURRENT
      CUMULATIVE MEAN
      CUM STANDARD DEV
      CUMULATIVE MIN
      CUMULATIVE MAX
      PERIOD MEAN
      PER STANDARD DEV
      PERIOD MIN
      PERIOD MAX
```

Figure 66. Options for Statistics

To calculate the current number of message Items in the Queue called "Buffer" at any given time, enter an "X" next to "CURRENT". The entities with respect to which graphs can be defined are Resources, Queues, Processes, Items and Variables. Up to ten such graphs may be defined per analysis session.

6.3 STARTING THE SIMULATION

Once the model is initialized and graphs are defined, the model may be executed through a simulation run. The execution of the model may be triggered either for the entire Scenario or for a specified number of periods, so that global Variables can be given new values different from those previously defined in the DUI.

The values of Constants and the initial values of global Variables may also be changed before a simulation exercise begins. The former option will be chosen in this example to investigate the effect of altering the time required to transmit or to process message Items. To begin the simulation, type

```
GO 1
```

This command indicates that the simulation is to be run for 1 of the simulation periods defined when the model was created in the DUI. When the simulation starts, a message will be displayed which provides the real time at which the simulation began. As the first simulation period completes, a subsequent message will be displayed showing the period number completed out of the total number of periods, the simulation time, and the real time when the period ended. Then the screen will offer the following message

```
END OF PERIOD
YOU MAY NOW ENTER COMMANDS
```

6.4 EDITING VARIABLES BETWEEN SIMULATION STAGES

To change the value of a variable, issue the appropriate [illegible] information as [illegible] the type of entity to be edited, [illegible] entity whose value is to be changed and (c) the new numerical value of the entity. The Variable Gamma2 formerly had the value of .002. To change it to .001, type

```
E V,gamma2,.001
```

The simulation may be continued for two more periods by typing

```
GO 2
```

When this stage of the simulation is completed, the value of Variables may be changed back to .002 by typing

```
E V,gamma2,.002
```

To command that the remainder of the Scenario be run through without further interruption, type the GO command without a numeric parameter, thus:

```
GO
```

If no mistakes were made in constructing the model that cause the simulation to abort, the computer will prompt, after some time, that the simulation is completed.

The plot produced by this run is shown in figure 67, and the output report *appears in Appendix B.*

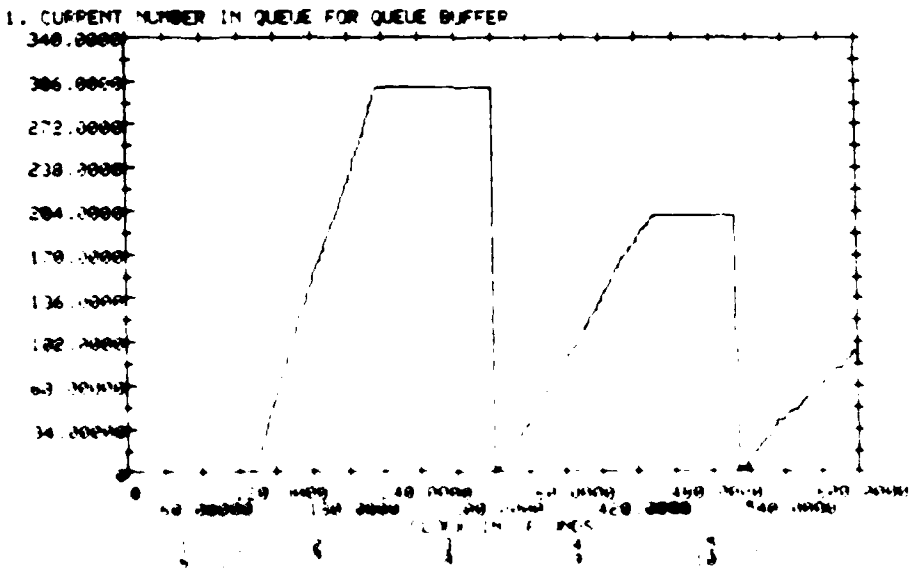


Figure 67. Plot from ... Example

7. A MORE ELABORATE EXAMPLE

In this section a communication system slightly more complicated than that designed in Section 5 and analyzed in Section 6 is constructed. To do this, however, requires that we introduce one further AISIM feature.

7.1 MESSAGE ROUTING SUBMODEL

When one Process is triggered by another through a CALL primitive, the called Process will execute in the same architectural node as the one that triggered it, i.e., utilize the same Resource, even if the two Processes are normally associated with different nodes. This is inconvenient in the representation of communication systems in which an activity in one hardware element causes activity in another one. AISIM therefore embodies a submodel to represent the situation in which a Process in one node triggers a Process in another node by communicating through the network architecture. This submodel consists of a collection of Processes and one Item.

The Processes that accomplish this must be placed in a project database with the commands available in the Library User Interface. The entities of this submodel need not be defined anew. For information on the use of this facility, see the AISIM User's Manual, Section 10.

7.2 DEFINING ARCHITECTURAL ELEMENTS

Consider modeling a communication system between two airbases, a headquarters and a command headquarters that communicates directly with a computer disk. Between these end-points are switches that govern the routing of messages through the system. The physical layout of this system is shown in figure 68.

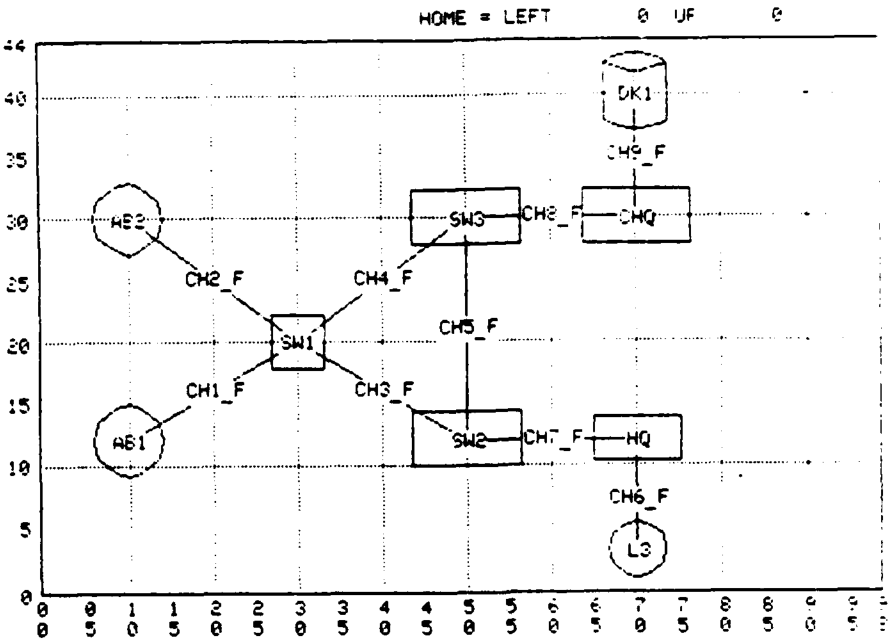


Figure 68. System Architecture

For this example, the shortest paths between the nodes will be used. Therefore, subsequent to defining the architecture, method B is used to create the Legal Path Table. The resulting table is depicted in the analysis report given in Appendix C.

The operations associated with this architecture are as follows. Both airbases periodically broadcast messages to the other nodes in the system and request plans from the command headquarters.

The effect of each broadcast is to (1) stimulate processing in the HQ and CHQ and to cause the updating of information in all other nodes. Periodically an applications program in L3 requests plans from the CHQ, as do AB1 and AB2. The effect of any such request is to engage the operation of the disk that communicates directly (and only) with the CHQ.

This description of the main operations of the system implies the following more rigorous listing of the Processes that need to be defined to represent such a system. The Processes required will be:

—A Process to represent the request from the HQ to the CHQ for plans. It will execute in the HQ node and will trigger a Process in the CHQ node.

--A Process to represent the broadcast of data from AB1 and AB2 to all other nodes. This Process will execute in the nodes AB1 and AB2 and will trigger an updating Process in (a) the HQ node, (b) the CHQ node and (c) each other, i.e., a broadcast in one airbase will update information in the other.

--A Process to represent the updating activity that occurs in the CHQ, triggered by broadcasts from the airbases.

—A Process to represent the updating activity that occurs in the HQ that is triggered by broadcasts from the airbases.

—A Process to represent the updating activity in the airbases which is triggered by broadcasts from one another.

—A Process to represent the formulation of plans at the CHQ, which is triggered by requests from AB1, AB2 and HQ. This Process executes in the CHQ node and triggers another Process representing disk operation in the disk node.

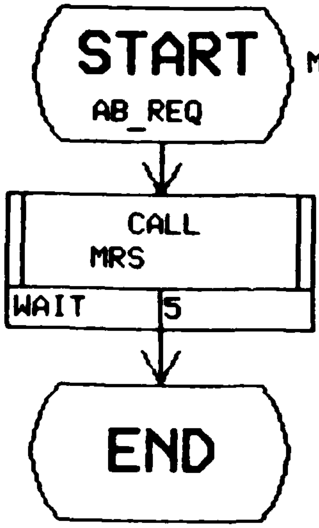
—A Process to represent the operation of the disk that communicates with the CHQ node.

These descriptions can be used to generate the AISIM Process definitions found on the following pages. The Process flowcharts for each are displayed, together with annotations to clarify the rationale for the steps that might otherwise be obscure.

AIRBASE REQUEST FOR PLANS REPORT FROM CHQ

GIVEN

MSG



MSG

RETURN

NO

The Process is given the value of the variable MSG which happens to be an ITEM.

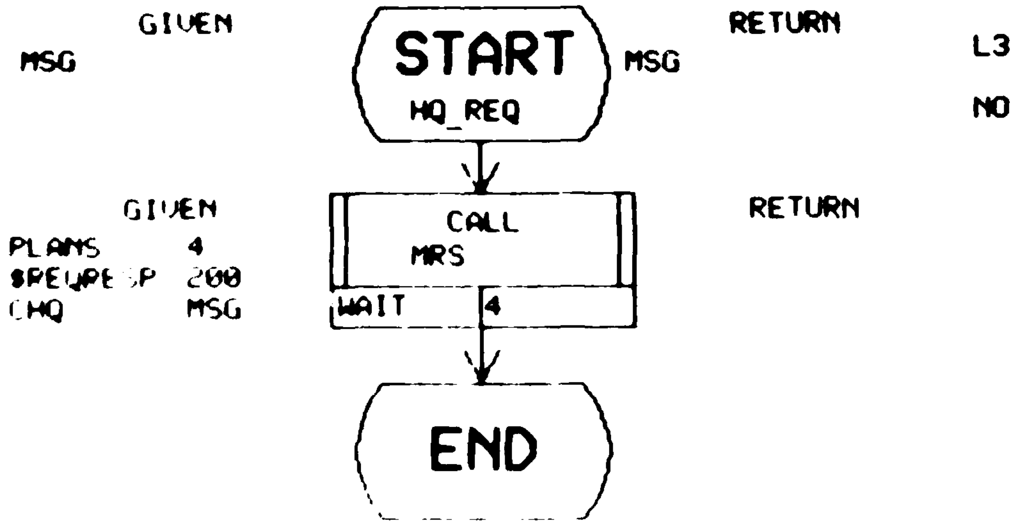
GIVEN

PLANS	5
$REQRESP	200
CHQ	MSG

RETURN

This primitive triggers a request for plans from the CHQ for an airbase.

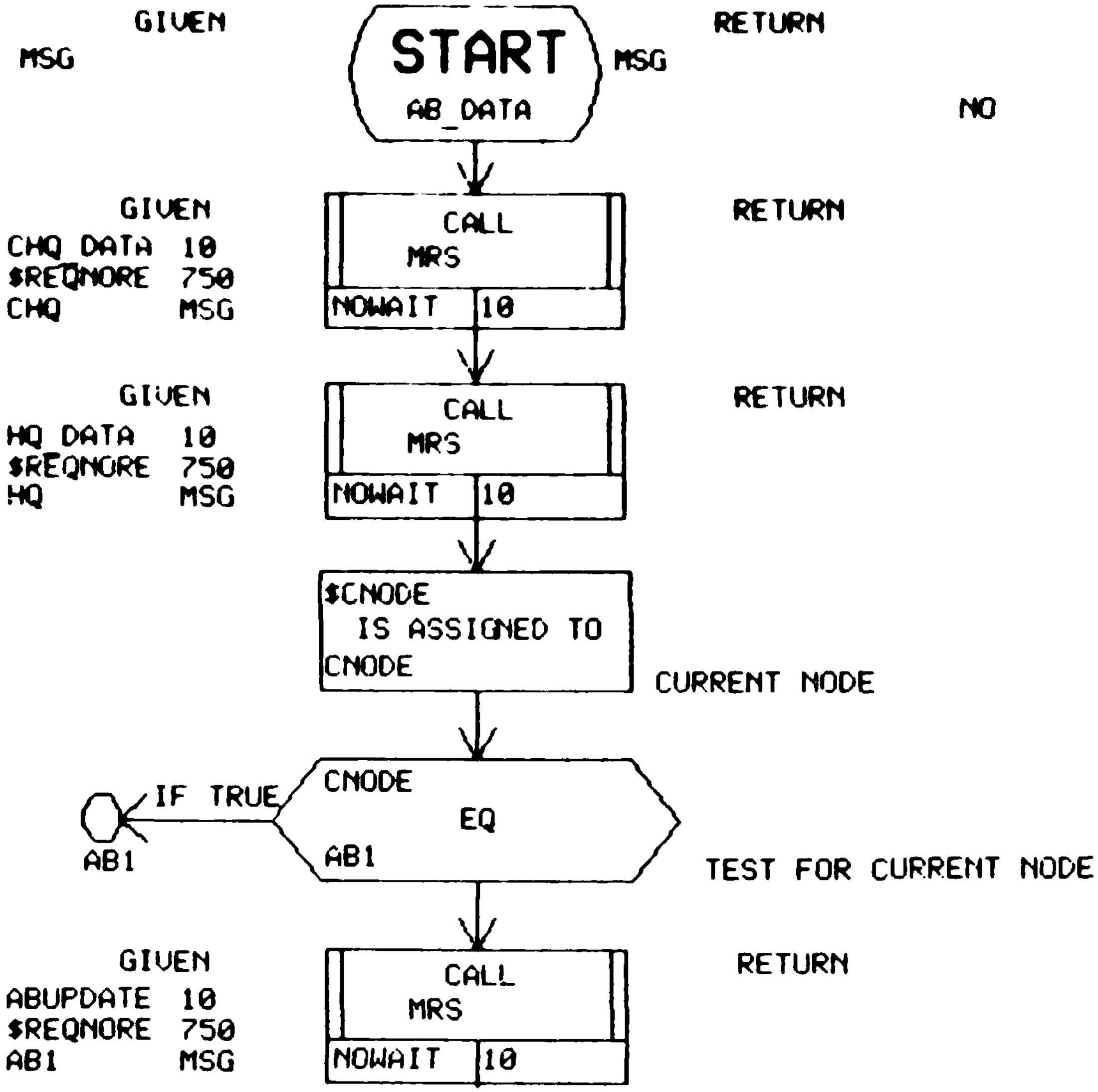
HQ REQUEST FOR STATUS DISPLAY FROM CHQ



The Process is given the value of the variable MSG which happens to be an Item.

This primitive triggers a request for plans from the CHQ for HQ.

AIR BASE STATUS BROWADCAST TO ALL OTHER NODES



The Process is given the value of a local variable "MSG" which in this case resolves to an Item representing a transient data element.

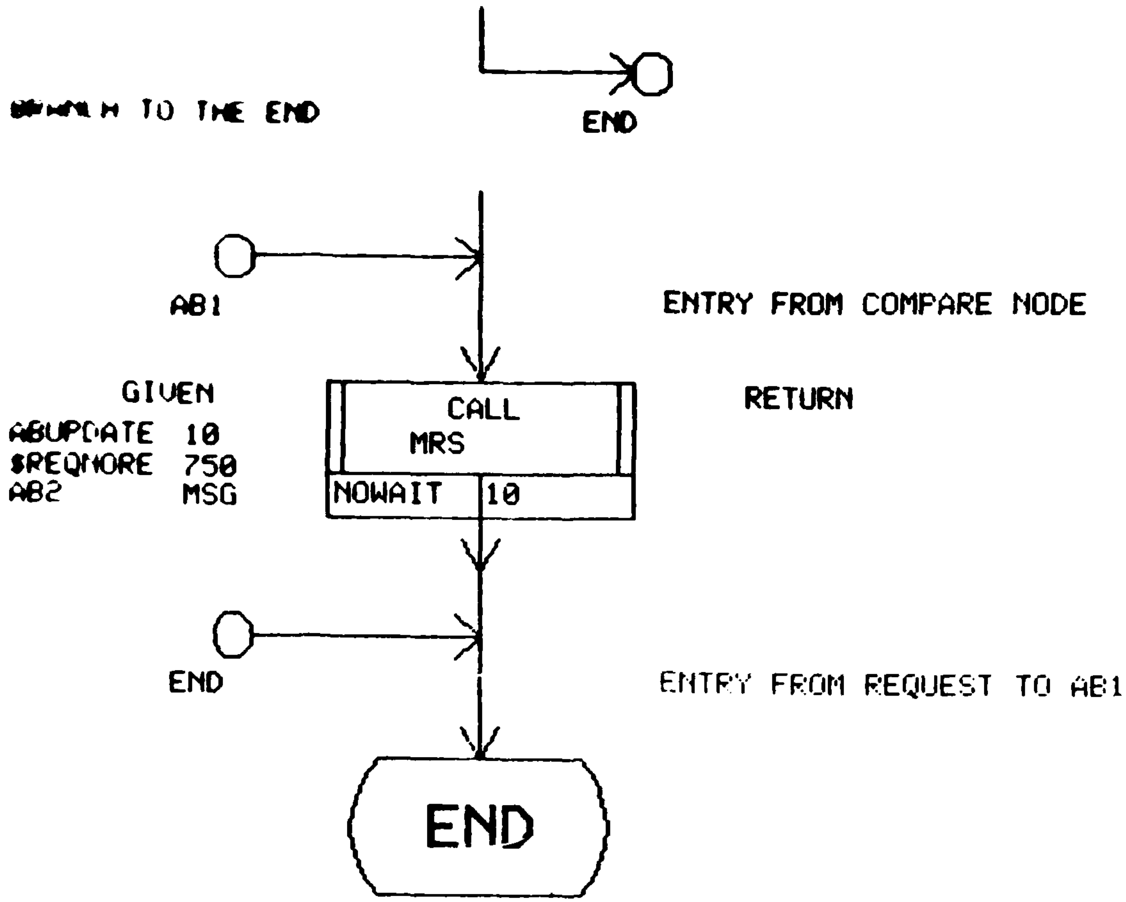
This primititive triggers a data update in the CHQ.

This primitive triggers a data update in the HQ.

The literal "$CNODE" is assigned to the keyword "CNODE" which denotes the Node in which the Process is currently triggering. "AB-DATA" takes place in two different Nodes, AB1 and AB2.

This primitive determines whether the Node in which the Process is presently executing is AB1 or AB2. If it is in AB1 it branches to the the Entry primitive labled "AB1" below.

This Call primitive triggers a data update in the AB1 Node.

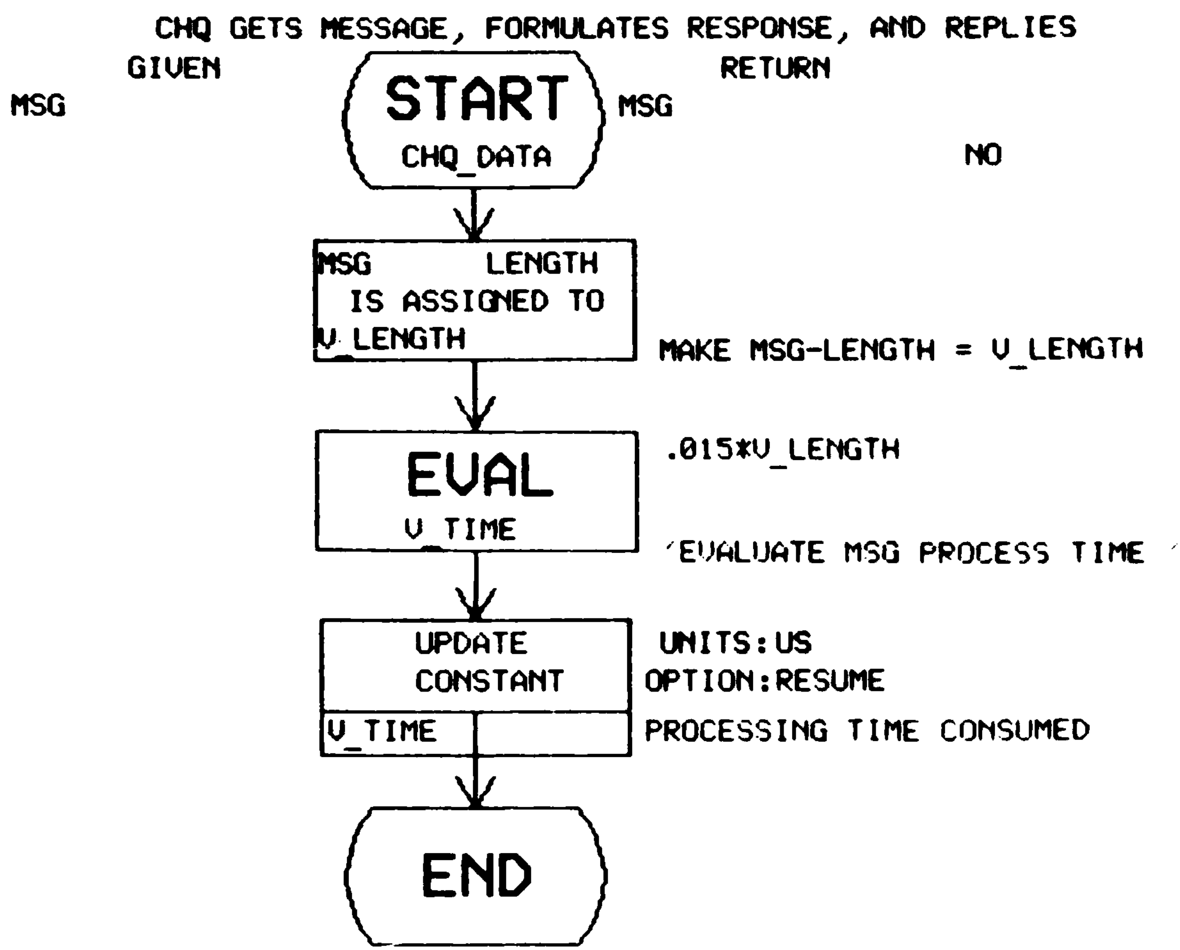


This primitive branches to the Entry primitive labled END below.

The process branches to this Entry primitive if the Process is executing in the AB1 Node.

This Call primitive triggers data updating in the AB2 Node.

The Process branches to this Entry primitive if the Process is executing in the AB2 Node.



The Process is given the value of a local variable "MSG" which in this case resolves to an Item representing a transient data element.

The attribute carried by the Item MSG called "LENGTH" is assigned to the variable "V_LENGTH".

The variable V_TIME which represents the time required to process the data from the airbases is calculated as (V_LENGTH) X (.015).

This Action primitive consumes time equal to the value of V_TIME calculated above.

HQ GETS MESSAGE, FORMULATES RESPONSE, AND REPLIES

GIVEN

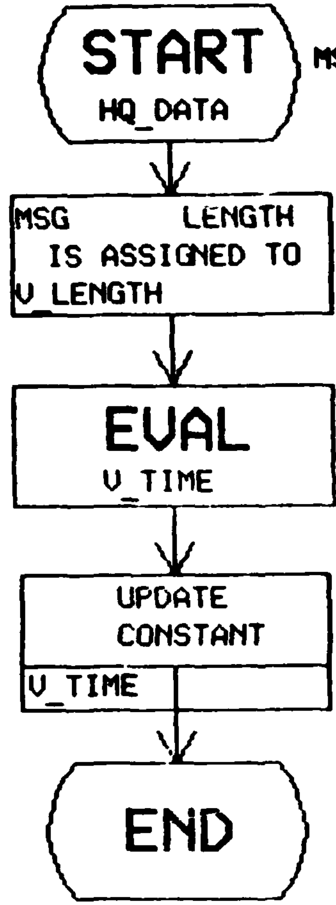
MSG

RETURN

MSG

NO

This Process differs from CHQ_DATA only in the Node in which it takes place.

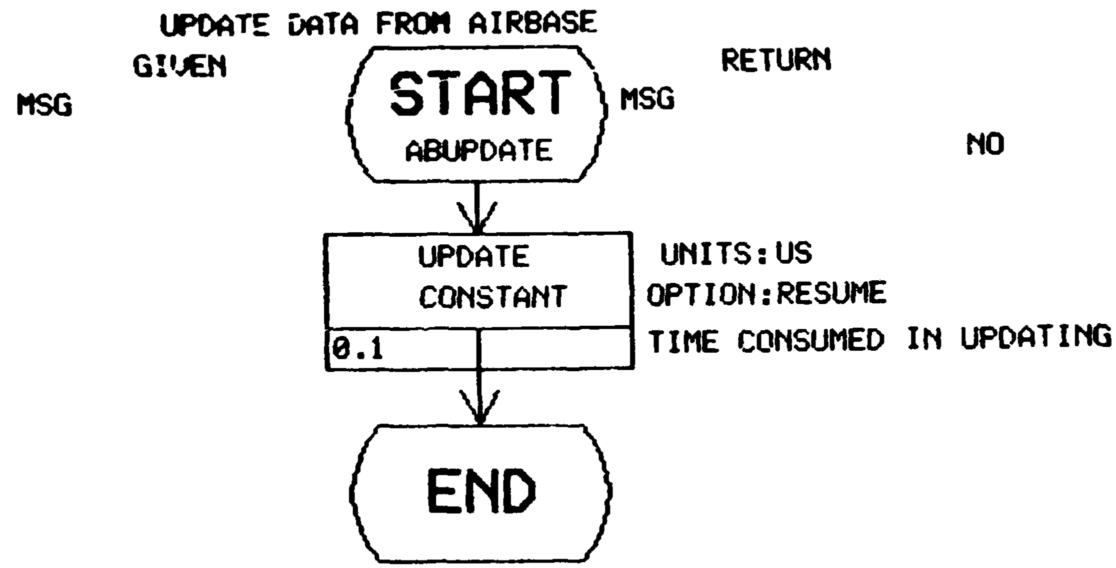


MAKE MSG_LENGTH = V_LENGTH

.015*V_LENGTH

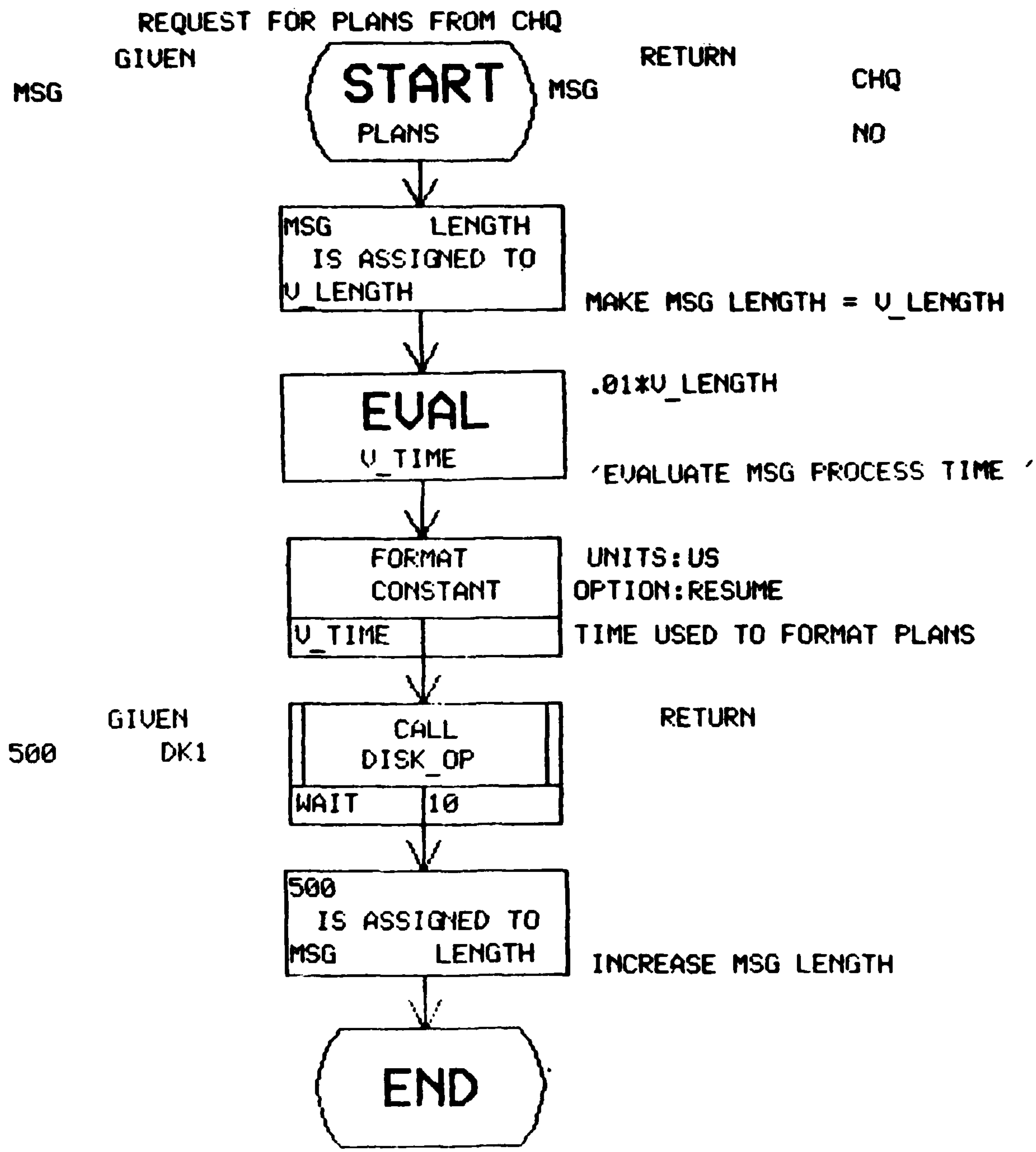
'EVALUATE MSG PROCESS TIME '

UNITS:US
OPTION:RESUME
PROCESSING TIME CONSUMED



This Process is given the value of a local variable "MSG" which in this case resolves to an Item representing a transient data element.

Consume 0.1 microseconds of time to perform update processing.



The Process is given the local variable MSG which denotes an Item.

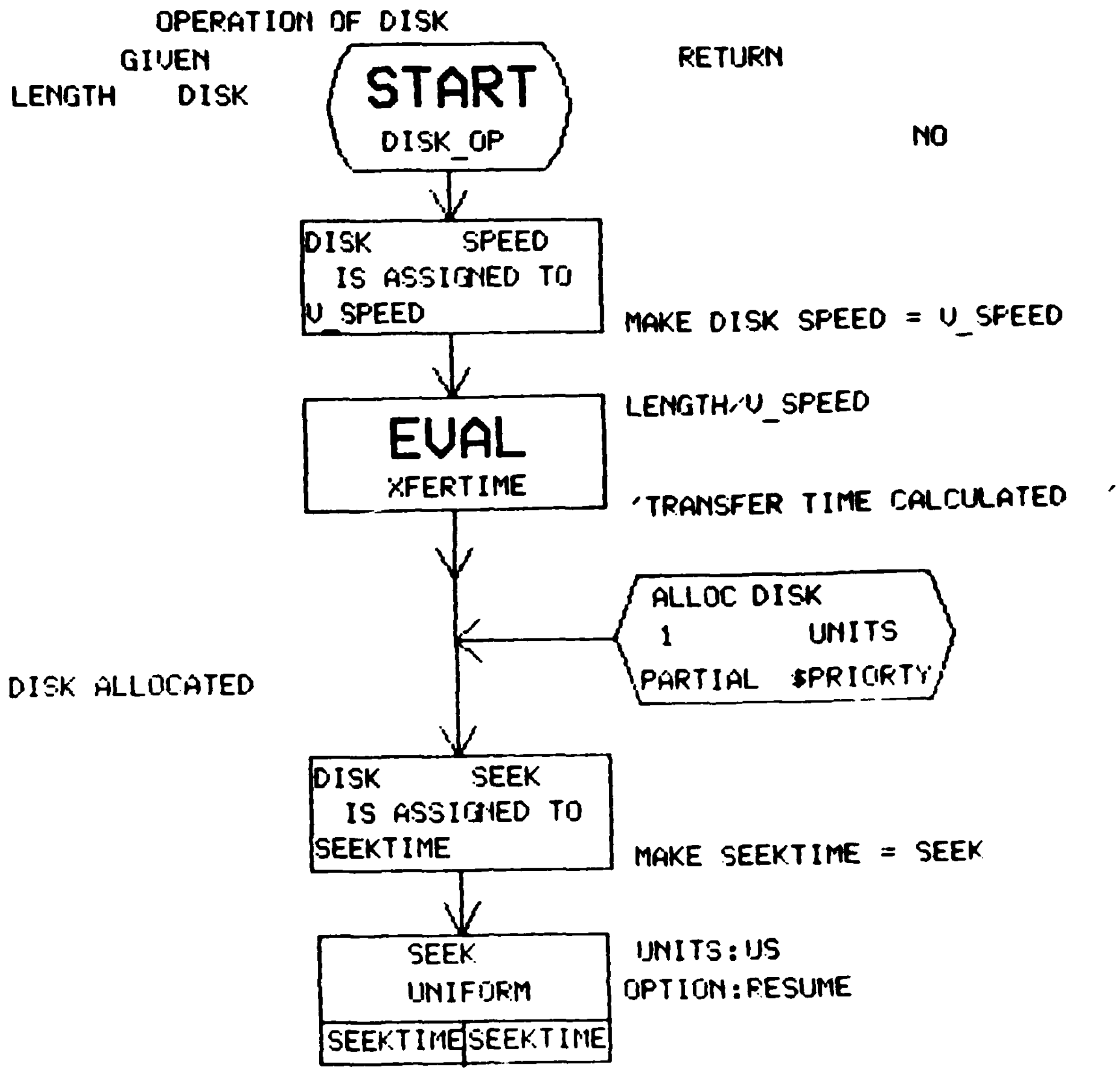
The attribute LENGTH of the Item MSG called is assigned to the local variable V_LENGTH.

The processing time is calculated as .01 times the message length.

The calculated amount of time is consumed for formatting plans.

The process to perform disk operations is invoked. PLANS does not continue until this process is completed.

This message length for the return message is changed to 500.



The Process is given the variables LENGTH and DISK which resolve, respectitively, to a numeric and a Resource.

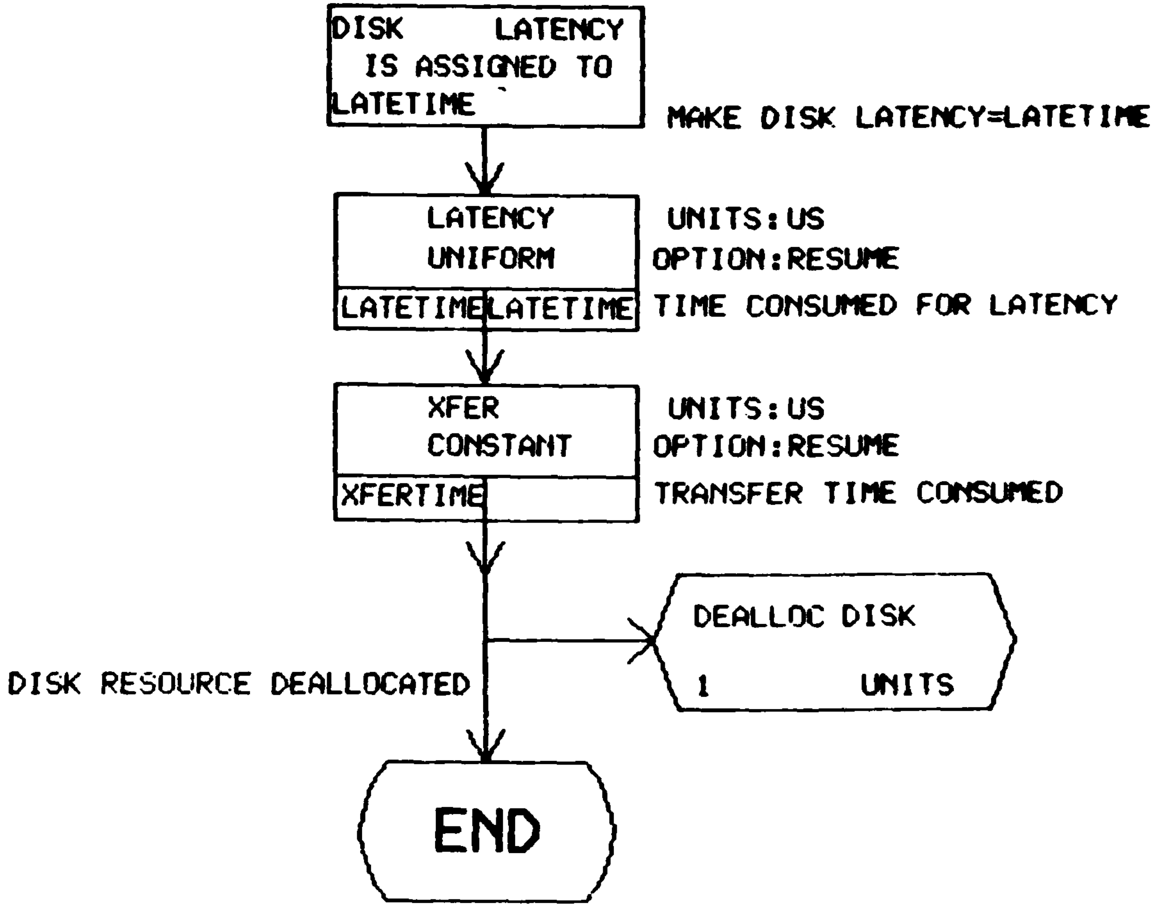
The attribute of the (Resource) DISK called SPEED is assigned to the local variable V_SPEED.

The transfer time is set equal to the length of the message divided by V_SPEED.

The Resource DISK is allocated.

The value of the disk attribute SEEK is assigned to the local variable SEEKTIME.

Time in the amount of SEEKTIME is taken up finding data on the disk.



The value of the attribute (of the Resource "DISK") is assigned to the local variable "LATETIME".

Time is consumed in the amount of LATETIME to represent latency.

Time is consumed to represent transfer time in the constant amount carried by the global constant XFERTIME.

The disk is deallocated.

[illegible] DEFINING REMAINING MODEL ELEMENTS

The remaining components of the model must be defined. These [illegible] Resources accessed by the Processes, [illegible] Actions which appear as Primitives, [illegible] all constants and global variables used in the Processes, [illegible] Loads, and [illegible] a Scenario.

[illegible] RESOURCE DEFINITIONS

All the Resources necessary to this model will have been defined automatically with default values while representing the physical layout represented in the Architecture Design Editor. Thus, if the nodes and links have the same names as in figure 68 above, the following list of Resources will already exist in the model database.

AB1	RESOURCE FOR NODE
AB2	RESOURCE FOR NODE
CH1_A	RESOURCE FOR CHANNEL CONNECTOR
CH1_B	RESOURCE FOR CHANNEL CONNECTOR
CH2_A	RESOURCE FOR CHANNEL CONNECTOR
CH2_B	RESOURCE FOR CHANNEL CONNECTOR
CH3_A	RESOURCE FOR CHANNEL CONNECTOR
CH3_B	RESOURCE FOR CHANNEL CONNECTOR
CH4_A	RESOURCE FOR CHANNEL CONNECTOR
CH4_B	RESOURCE FOR CHANNEL CONNECTOR
CH5_A	RESOURCE FOR CHANNEL CONNECTOR
CH5_B	RESOURCE FOR CHANNEL CONNECTOR
CH6_A	RESOURCE FOR CHANNEL CONNECTOR
CH6_B	RESOURCE FOR CHANNEL CONNECTOR
CH7_A	RESOURCE FOR CHANNEL CONNECTOR
CH7_B	RESOURCE FOR CHANNEL CONNECTOR
CH8_A	RESOURCE FOR CHANNEL CONNECTOR
CH8_B	RESOURCE FOR CHANNEL CONNECTOR
CH9_A	RESOURCE FOR CHANNEL CONNECTOR
CH9_B	RESOURCE FOR CHANNEL CONNECTOR
CHQ	COMMAND HEAD-QUARTERS
DK1	DISK FOR COMMAND HEAD-QUARTERS
HQ	HEAD-QUARTERS
L3	RESOURCE FOR NODE
SW1	SWITCH BETWEEN AIRBASES AND OTHER TWO SWITCHES (1&2)
SW2	SWITCH BETWEEN SWITCH 1 & 3 AND HQ
SW3	SWITCH BETWEEN SWITCH 1 & 2 AND CHQ

Figure 69. Defined Resource Entities

Since these Resources are created with default values (an initial unit of 1, a maximum unit of 1, and a default description), they must be edited to provide helpful descriptions and to give them attributes since attributes of these Resources are accessed in several places in the Message Routing Submodel. Descriptions and attributes can be edited before generating the architecture using the ADE's DEFINE command. See Section 2.2 of this manual.

7.3.2 FILLING IN THE ACTION DEFINITIONS

The Action Primitives in the Processes must have corresponding Action entity definitions outside the Process which are automatically created by AISIM. The Actions used are these:

ACTION	DESCRIPTION
FORMAT	TIME USED TO FORMAT PLANS FROM CHQ
LATENCY	LATENCY PAUSE SUBSEQUENT TO SEEK
ROUTE_ON	PROCESSING DELAY TO ROUTE A MESSAGE
SEEK	SEEKING INFORMATION ON DISK
UPDATE	UPDATING INFO SINCE PREVIOUS BROADCAST TO OTHER NODES
XFER	TRANSFER INFORMATION SOUGHT ON DISK
XFER_ON	PROCESSING DELAY TO ROUTE A MESSAGE OVER A CHANNEL

Figure 70. Defined Action Entities

7.3.3 CONSTANTS AND GLOBAL VARIABLES

This model contains five global Variables (ABDRATE, ABRATE, HQRATE, TIME1 and VRATE) and one Constant (V_TRACE). Their defined values and descriptions (which explain their role in the model) are as shown in figure 71.

ABDRATE	INTERVAL RATE BETWEEN SIGNALS
ABRRATE	INTERVAL RATE BETWEEN SIGNALS
HQRRATE	INTERVAL BETWEEN SIGNALS
TIME1	AVERAGE SEEK TIME FOR DISK IN MILLISECONDS
VRATE	SWITCH-OTHER NODE CHANNEL SPEED IN MS/BYTE
V_TRACE	DEFAULT IS NO TRACE ON

Figure 71. Defined Constant and Variable Entities

7.3.4 DEFINING LOADS AND SCENARIOS

In this model we wish to represent the several Process triggerings that are due to causes outside the system. First, the AB1 and AB2 will broadcast communications to the other nodes (which trigger updating Processes in them) every minute, by an interval scheduling method. In addition, AB1 and AB2 will issue requests for plans from the CHQ sixty times in one hour by an exponential scheduling method. We define a second Load to represent requests from the leaf-node, L3, also for plans from the CHQ. This Process will also be undertaken sixty times per hour, exponentially distributed. The Load definitions implied by these requirements are printed in appendix C.

The length of the entire Scenario is 360,000 milliseconds (one hour), which is divided into ten periods of six minutes each. To simulate the operation of the system with the worst case, we stipulate that both of the functional Loads are triggered simultaneously, at the beginning of the Scenario. In addition, as a monitoring device, we initiate the Trace Process at the beginning of the simulation run. The parameters for the Scenario implied by these requirements are printed page 16 of the analysis report in appendix C.

7.4 ANALYZING THE MODEL

To run the model through a simulation test, invoke the Analysis User Interface from the AISIM READY level. For this example, the simulation will not be interrupted at the ends of periods, nor will graphs be defined.

The analysis report obtained from a simulation run of this model appears in appendix C.

APPENDIX A

TERMINAL PROFILES FOR FORMS

	UP	DOWN	LEFT	RIGHT	ENTER	+FIELD	-FIELD
HP2647A	F1	F2	F3	F4	F5	<cr>	F6
HP2648A	F1	F2	F3	F4	F5	<cr>	F6
HP2623	F1	F2	F3	F4	F5	<cr>	F6
TEK4105	F1	F2	F3	F4	F5	<cr>	F6
VT100	↑	↓	<—	-->	PF1	<cr>	PF2

Figure 72. Terminal Profiles for Forms

Figure 72 describes the function keys which are used to move through the forms on each terminal. Following is a description of the ways in which a user can move through a form. These movements correspond to the column headings in the figure.

UP - If the cursor is in a block of fields, such as Resource attributes, the cursor will move up to the field above it. If the cursor is in a single field or at the top of a block, the cursor will move to the end of the next field above it. If there are no fields above it, the cursor will wrap to the end of the last field in the form.

DOWN - If the cursor is in a block of fields, such as Resource attributes, the cursor will move down to the field below it. If the cursor is in a single field or at the bottom of a block, the cursor will move to the beginning of the next field below it. If there are no fields below it, the cursor will wrap to the beginning of the first field in the form.

LEFT - The cursor will move one position to the left in the current field. If the cursor is at the beginning of a field, it will move to the end of the previous field. If the cursor is at the top of the form, it will wrap to the end of the last field in the form.

RIGHT - The cursor will move one position to the right in the current field. If the cursor is at the end of a field, it will move to the beginning of the next field. If the cursor is at the end of the form, it will wrap to the beginning of the first field in the form.

ENTER - Exit the form and send the data in the form to be processed by the AISIM system.

+FIELD - Move the cursor to the beginning of the next field in the form. If the cursor is at the end of the form, it will wrap to the top of the form.

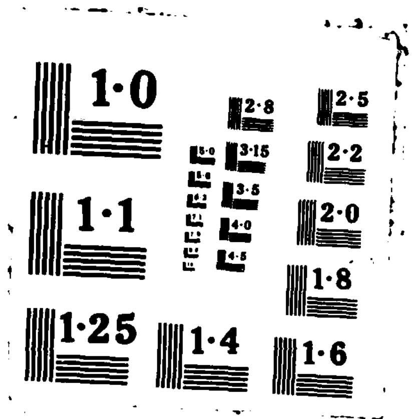
-FIELD - Move the cursor to the end of the previous field in the form. If the cursor is at the top of the form, it will wrap to the end of the last field in the form.

APPENDIX B

SIMULATION REPORT FOR WORKING EXAMPLE

AD-A188 998 AUTOMATED INTERACTIVE SIMULATION MODEL (AISIM) VAX VERSION 50 TRAINING MA..(U) HUGHES AIRCRAFT CO FULLERTON CA GROUND SYSTEMS GROUP V ALLERTON ET AL. 2/2

UNCLASSIFIED 29 MAY 87 HAC-1854896-2 ESD-TR-87-232 F/G 12/5 NL


1·0
2·8
2·5
3·15
2·2
3·5
1·1
2·0
4·0
4·5
1·8
1·25
1·4
1·6

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$                    S I M U L A T I O N   R E P O R T                  $
$                           AISIM VERSION 6.0                           $
$                        HUGHES AIRCRAFT COMPANY                        $
$                               05/15/87                                $
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

                      06/01/1987          13:47:18

GLOBAL CONSTANT DEFINITION.....

CONSTANT INITIAL
MNEMONIC VALUE    COMMENT
======== ======== ==================================================

FILE DEFINITION.....

FILE
MNEMONIC          COMMENT
======== ======== ==================================================

TABLE DEFINITION....


GLOBAL VARIABLE DEFINITION.....

VARIABLE INITIAL
MNEMONIC VALUE    COMMENT
======== ======== ==================================================
GAMMA1   700
GAMMA2   .002

ITEM DEFINITION......


ITEM                       DESCRIPTION
======== ================= ==========================================
MSG                        MSG TO BE TRANSMITTED
         ATTR.    INITIAL
         NAME     VALUE
         ======== ========
         LENGTH   $LENGTH


QUEUE DEFINITION.....

QUEUE    MAXIMUM
MNEMONIC   SIZE   COMMENT
```

```
PAGE     2
======== ======== ================================================
BUFFER   INFINITE BUFFER ON WHICH MESSAGES ARE STORED

RESOURCE DEFINITION.....

RESOURCE TOTAL    INITIAL
MNEMONIC # UNITS  # UNITS  DESCRIPTION
======== ======== ======== =======================================
BUF1     1        1        RESOURCE ASSOCIATED WITH BUFFER

ARCHITECTURE LEGAL PATH DEFINITION

FROM      TO        NEXT      VIA
DEVICE    DEVICE    DEVICE    LINK
========  ========  ========  ========

ACTION DEFINITION.....

ACTION
MNEMONIC          COMMENT
======== ======== ================================================
READ MSG          READ A MESSAGE
SENDING           TRANSMIT A MESSAGE

PROCESS DEFINITION.....

PROCESS
MNEMONIC          DESCRIPTION
======== ======== ================================================
RECEIVE           RECEIVE MESSAGES FROM TRANSMIT

ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
======== ======== ======== ======== ======== =======================
         START             NO
         TEST     BUF1     ABORT             TEST FOR BUFFER USE
         REMOVE   FIRST    MSG      BUFFER   REMOVE BY FIFO DISCIPLINE
         COMPARE  MSG               EQ       WHEN MSG=Ø BUFFER IS EMPTY
                  Ø                 ABORT
         ASSIGN   MSG      LENGTH            MSG LENGTH IS READ
                  ALPHA
         EVAL     MU                         CALCULATE RECEPTION TIME
                  ALPHA*GAMMA2
         READ_MSG CONSTANT MU                TIME TO PROCESS MESSAGE
                  SECONDS  RESUME
         DESTROY  MSG                        MSG ELIMINATED FROM SYSTEM
ABORT    ENTRY                               ENTER FROM COMPARE & TEST
         END
```

```
PAGE    3
LOCAL VARIABLES OF PROCESS RECEIVE
=======================================================================
    1 BUF1        (R)     2 MSG         (I)     3 BUFFER     (Q)     4 ALPHA
    5 MU                  6 READ_MSG    (A)

GLOBAL VARIABLES OF PROCESS RECEIVE
=======================================================================
    1 GAMMA2
PROCESS
MNEMONIC           DESCRIPTION
======== ======== =====================================================
TRANSMIT           TRANSMITTING MESSAGES TO RECEIVER


ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
======== ======== ======== ======== ======== ==========================
         START             NO
         ALLOC    BUF1     1        ALL      ALLOCATE BUF1
                  $PRIORTY
         CREATE   MSG                        INTRODUCE MSG INTO SYSTEM
         EVAL     ALPHA                      TWICE AVERAGE * RANDOM NUM
                  RANDOM*GAMMA1
         ASSIGN   ALPHA                      SET MESSAGE LENGTH
                  MSG      LENGTH
         EVAL     MU                         CALCULATE TRANSMIT TIME
                  ALPHA*GAMMA2
         SENDING  CONSTANT MU                TIME CONSUMED TRANSMITTING
                  SECONDS  RESUME
         FILE     MSG      LAST     BUFFER   STORE MSG ON BUFFER
         DEALLOC  BUF1     1                 RELEASE RESOURCE BUF1
         END

LOCAL VARIABLES OF PROCESS TRANSMIT
=======================================================================
    1 BUF1        (R)     2 MSG         (I)     3 ALPHA              4 MU
    5 SENDING     (A)     6 BUFFER      (Q)

GLOBAL VARIABLES OF PROCESS TRANSMIT
=======================================================================
    1 GAMMA1              2 GAMMA2

LOAD DEFINITION.....

LOAD
MNEMONIC           DESCRIPTION
======== ======== =====================================================
L1
         LOAD     NODES
         ======== ======== ======== ========
```

```
         PROCESS           SCHEDULE
         MNEMONIC MAX #    METHOD   MEAN     DELTA    UNITS    PRIORITY
         ======== ======== ======== ======== ======== ======== ========
         TRANSMIT 300      POISSON                    SECONDS  0
LOAD
MNEMONIC          DESCRIPTION
======== ======== ================================================
L11
         LOAD     NODES
         ======== ======== ======== ========


         PROCESS           SCHEDULE
         MNEMONIC MAX #    METHOD   MEAN     DELTA    UNITS    PRIORITY
         ======== ======== ======== ======== ======== ======== ========
         RECEIVE  600      INTERVAL 0.00001           SECONDS  0
LOAD
MNEMONIC          DESCRIPTION
======== ======== ================================================
L2
         LOAD     NODES
         ======== ======== ======== ========


         PROCESS           SCHEDULE
         MNEMONIC MAX #    METHOD   MEAN     DELTA    UNITS    PRIORITY
         ======== ======== ======== ======== ======== ======== ========
         TRANSMIT 200      POISSON                    SECONDS  0
LOAD
MNEMONIC          DESCRIPTION
======== ======== ================================================
L22
         LOAD     NODES
         ======== ======== ======== ========


         PROCESS           SCHEDULE
         MNEMONIC MAX #    METHOD   MEAN     DELTA    UNITS    PRIORITY
         ======== ======== ======== ======== ======== ======== ========
         RECEIVE  400      INTERVAL 0.00001           SECONDS  0
LOAD
MNEMONIC          DESCRIPTION
======== ======== ================================================
```

```
PAGE     5
L3
         LOAD     NODES
         ======== ======== ======== ========


         PROCESS           SCHEDULE
         MNEMONIC MAX #    METHOD   MEAN     DELTA    UNITS    PRIORITY
         ======== ======== ======== ======== ======== ======== ========
         TRANSMIT 100      POISSON                    SECONDS  0
LOAD
MNEMONIC          DESCRIPTION
======== ======== ==============================================
L33
         LOAD     NODES
         ======== ======== ======== ========


         PROCESS           SCHEDULE
         MNEMONIC MAX #    METHOD   MEAN     DELTA    UNITS    PRIORITY
         ======== ======== ======== ======== ======== ======== ========
         RECEIVE  200      INTERVAL 0.00001           SECONDS  0

SCENARIO DEFINITION....

SCENARIO
MNEMONIC          DESCRIPTION
======== ======== ==============================================
SCEN

PERIOD   PERIOD   OUTPUT
LENGTH   UNITS    UNITS
======== ======== ========
100      SECONDS  SECONDS

         PERIOD   PERIOD   PERIOD   PERIOD   PERIOD   PERIOD   PERIOD
         MNEMONIC MNEMONIC MNEMONIC MNEMONIC MNEMONIC MNEMONIC MNEMONIC
======== ======== ======== ======== ======== ======== ======== ========
         1        2        3        4        5        6

TRIGGER  TIME  TO SCHEDULE SCHEDULE TRIGGER  TIME  TO SCHEDULE SCHEDULE
MNEMONIC SCHEDULE UNITS    PRIORITY MNEMONIC SCHEDULE UNITS    PRIORITY
======== ======== ======== ======== ======== ======== ======== ========
L1       100      SECONDS  0        L11      100      SECONDS  0
L2       300      SECONDS  0        L22      300      SECONDS  0
L3       500      SECONDS  0        L33      500      SECONDS  0
```

```
PAGE    6
####    0 ERRORS WERE DETECTED DURING MODEL INITIALIZATION
```

```
PAGE    8

SIMULATION TIME =        600.00000 SECONDS

VARIABLE REPORT

NUMERIC VARIABLES...

         TOTAL    -------------------------VALUE-------------------------
VARIABLE SAMPLES. CURRENT... MEAN...... STD DEV... MINIMUM... MAXIMUM...
======== ======== ========== ========== ========== ========== ==========
GAMMA1          1    700.000    700.000         0.    700.000    700.000
GAMMA2          3      0.002      0.002 +4.714E-04      0.001      0.002

NON-NUMERIC VARIABLES...

         CURRENT  CURRENT
VARIABLE TYPE     VALUE
======== ======== ========
```

PAGE 9

SIMULATION TIME = 600.00000 SECONDS

ITEM REPORT

ITEM NAME	NUMBER CREATED	NUMBER DESTR'D	TIME IN SYSTEM MINIMUM...	MAXIMUM...	AVERAGE...	STD DEV...
MSG	596	500	70.50	200.88	142.90	33.70

SIMULATION TIME = 600.00000 SECONDS

QUEUE REPORT

QUEUE	TOTAL NUMBER	CURRENT...	MEAN......	STD DEV...	MINIMUM...	MAXIMUM...
BUFFER						
FILED ON	596					
REMOVED FROM	500					
# IN QUEUE		96.000	126.480	107.531	0.	300.000
TIME IN QUEUE			141.755	33.713	67.932	199.664
TASKS BLOCKED	0					
TASKS RESUMED	0					
# BEING BLOCKED		0.	0.	0.	0.	0.
TIME BLOCKED			0.	0.	0.	0.

PAGE 11

SIMULATION TIME = 600.00000 SECONDS

RESOURCE REPORT

RESOURCE	TOTAL NUMBER	CURRENT...	MEAN......	STD DEV...	MINIMUM...	MAXIMUM...
BUF1						
# IDLE		1.000	0.513	0.500	0.	1.000
REQUEST TIME			7.171	6.273	0.	24.357
HOLD TIME	0		0.	0.	0.	0.
INTO BUSY	596					
OUT OF BUSY	596					
# BUSY		0.	0.487	0.500	0.	1.000
BUSY TIME			0.490	0.348	+3.510E-04	1.398
# INACTIVE		0.	0.	0.	0.	0.
INTO WAIT	596					
OUT OF WAIT	596					
# WAITING		0.	7.123	10.523	0.	36.000
WAIT TIME			7.171	6.273	0.	24.357

CURRENTLY ALLOCATED
TO PROCESSES: NONE

PROCESSES CURRENTLY
WAITING: NONE

SIMULATION TIME = 600.00000 SECONDS

ACTION REPORT

ACTION	TOTAL SAMPLES	MEAN.....	STD DEV...	MINIMUM..	MAXIMUM...	% TIME OF TOTAL.
READ MSG						
USEFUL TIME	500	0.675	0.410	+7.019E-04	1.398	56.244
DELAY TIME	500	0.	0.	0.	0.	
WASTED TIME	0	0.	0.	0.	0.	

ACTION	TOTAL SAMPLES	MEAN.....	STD DEV...	MINIMUM..	MAXIMUM...	% TIME OF TOTAL.
SENDING						
USEFUL TIME	596	0.490	0.348	+3.510E-04	1.398	48.722
DELAY TIME	596	0.	0.	0.	0.	
WASTED TIME	0	0.	0.	0.	0.	

```
SIMULATION TIME =        600.00000 SECONDS

PROCESS REPORT

                    TOTAL
PROCESS             SAMPLES. SUM....... MEAN...... STD DEV... MINIMUM... MAXIMUM...
=============== ======== ========== ========== ========== ========== ==========
RECEIVE
          TOTAL     1200    337.465      0.281      0.425      0.         1.398
   PROCESS WAIT        0      0.         0.         0.         0.         0.
  RESOURCE WAIT        0      0.         0.         0.         0.         0.

        TOTAL #  # AUTO   # CALL   # OF     # NOT    # TIMES
        SCHEDULE SCHEDULE SCHEDULE COMPLETE COMPLETE SUSPEND.
        ======== ======== ======== ======== ======== ========
           1200     1200        0     1200        0        0

         ITEM     CREATED  RECEIVED SENT     DESTR'D
         ======== ======== ======== ======== ========
         MSG             0        0        0      500

                  PROCESS  HOLDING  TIME
         ITEM     # SMPLS  MEAN...... MINIMUM..  MAXIMUM... STD DEV...
         ======== ======== ========== ========== ========== ==========
         MSG           500       0.67 +7.019E-04       1.40       0.41

PROCESS           DESCRIPTION
======== ======== ==================================================
RECEIVE           RECEIVE MESSAGES FROM TRANSMIT

COUNT ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
===== ======== ======== ======== ======== ======== ========================
 1200          START             NO
 1200          TEST     BUF1     ABORT             TEST FOR BUFFER USE
 1200          REMOVE   FIRST    MSG      BUFFER   REMOVE BY FIFO DISCIPLINE
 1200          COMPARE  MSG               EQ       WHEN MSG=0 BUFFER IS EMPTY
 1200                   0                 ABORT
  500          ASSIGN   MSG      LENGTH            MSG LENGTH IS READ
  500                   ALPHA
  500          EVAL     MU                         CALCULATE RECEPTION TIME
  500                   ALPHA*GAMMA2
  500          READ_MSG CONSTANT MU                TIME TO PROCESS MESSAGE
  500                   SECONDS  RESUME
  500          DESTROY  MSG                        MSG ELIMINATED FROM SYSTEM
 1200 ABORT    ENTRY                               ENTER FROM COMPARE & TEST
 1200          END

                 TOTAL
```

PAGE 14

PROCESS	SAMPLES.	SUM.......	MEAN......	STD DEV...	MINIMUM...	MAXIMUM...
TRANSMIT						
TOTAL	596	4566.214	7.661	6.335	0.008	25.420
PROCESS WAIT	0	0.	0.	0.	0.	0.
RESOURCE WAIT	596	4273.884	7.171	6.273	0.	24.357

TOTAL # SCHEDULE	# AUTO SCHEDULE	# CALL SCHEDULE	# OF COMPLETE	# NOT COMPLETE	# TIMES SUSPEND.
596	596	0	596	0	0

ITEM	CREATED	RECEIVED	SENT	DESTR'D
MSG	596	0	0	0

ITEM	PROCESS # SMPLS	HOLDING MEAN......	TIME MINIMUM..	MAXIMUM...	STD DEV...
MSG	596	0.49	+3.510E-04	1.40	0.35

PROCESS		DESCRIPTION
TRANSMIT		TRANSMITTING MESSAGES TO RECEIVER

COUNT	ENTRY	OPCODE	PARM	PARM	PARM	COMMENT
596		START		NO		
596		ALLOC	BUF1	1	ALL	ALLOCATE BUF1
596			$PRIORTY			
596		CREATE	MSG			INTRODUCE MSG INTO SYSTEM
596		EVAL	ALPHA			TWICE AVERAGE * RANDOM NUM
596			RANDOM*GAMMA1			
596		ASSIGN	ALPHA			SET MESSAGE LENGTH
596			MSG	LENGTH		
596		EVAL	MU			CALCULATE TRANSMIT TIME
596			ALPHA*GAMMA2			
596		SENDING	CONSTANT	MU		TIME CONSUMED TRANSMITTING
596			SECONDS	RESUME		
596		FILE	MSG	LAST	BUFFER	STORE MSG ON BUFFER
596		DEALLOC	BUF1	1		RELEASE RESOURCE BUF1
596		END				

APPENDIX C

SIMULATION REPORT FOR ELABORATE EXAMPLE

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$                  S I M U L A T I O N   R E P O R T                 $
$                        AISIM VERSION 5.0                           $
$                    HUGHES AIRCRAFT COMPANY                         $
$                            05/15/87                                $
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

                      05/13/1987         09:06:06

GLOBAL CONSTANT DEFINITION.....

CONSTANT INITIAL
MNEMONIC VALUE    COMMENT
======== ======== ====================================================
V_TRACE  0        DEFAULT IS NO TRACE ON

FILE DEFINITION.....

FILE
MNEMONIC          COMMENT
======== ======== ====================================================

TABLE DEFINITION....


GLOBAL VARIABLE DEFINITION.....

VARIABLE INITIAL
MNEMONIC VALUE    COMMENT
======== ======== ====================================================
ABDRATE  60000    INTERVAL RATE BETWEEN SIGNALS
ABRRATE  36000    INTERVAL RATE  BETWEEN SIGNALS
HQRRATE  72000    INTERVAL BETWEEN SIGNALS
TIME1    30       AVERAGE SEEK TIME FOR DISK IN MILLISECONDS
VRATE    1.6276   SWITCH-OTHER NODE CHANNEL SPEED IN MS/BYTE

ITEM DEFINITION......


ITEM                       DESCRIPTION
======== ================= ============================================
MSG                        MESSAGE FOR INTERNODE COMMUNICATION
         ATTR.    INITIAL
         NAME     VALUE
         ======== ========
         CNODE    $CNODE
         FNODE    $CNODE
         LENGTH   99999999
```

```
PAGE     2
          RPROC     $ERROR
          RPROCPRI 99999999
          TNODE     $CNODE
          TYPE      $REQNORE

QUEUE DEFINITION.....

QUEUE     MAXIMUM
MNEMONIC    SIZE   COMMENT
======== ======== ================================================

RESOURCE DEFINITION.....

RESOURCE TOTAL    INITIAL
MNEMONIC # UNITS  # UNITS  DESCRIPTION
======== ======== ======== ========================================
AB1      1        1        RESOURCE FOR NODE
         ATTR.    INITIAL
         NAME     VALUE
         ======== ========
         M_ROUTE  8

AB2      1        1        RESOURCE FOR NODE
         ATTR.    INITIAL
         NAME     VALUE
         ======== ========
         M_ROUTE  8

CH1_A    1        1        RESOURCE FOR CHANNEL CONNECTOR
         ATTR.    INITIAL
         NAME     VALUE
         ======== ========
         RATE     VRATE

CH1_B    1        1        RESOURCE FOR CHANNEL CONNECTOR
         ATTR.    INITIAL
         NAME     VALUE
         ======== ========
         RATE     VRATE

CH2_A    1        1        RESOURCE FOR CHANNEL CONNECTOR
         ATTR.    INITIAL
         NAME     VALUE
         ======== ========
         RATE     VRATE

CH2_B    1        1        RESOURCE FOR CHANNEL CONNECTOR
         ATTR.    INITIAL
```

```
PAGE      3
          NAME      VALUE
          ========  ========
          RATE      VRATE

CH3_A     1         1         RESOURCE FOR CHANNEL CONNECTOR
          ATTR.     INITIAL
          NAME      VALUE
          ========  ========
          RATE      0.4069

CH3_B     1         1         RESOURCE FOR CHANNEL CONNECTOR
          ATTR.     INITIAL
          NAME      VALUE
          ========  ========
          RATE      0.4069

CH4_A     1         1         RESOURCE FOR CHANNEL CONNECTOR
          ATTR.     INITIAL
          NAME      VALUE
          ========  ========
          RATE      0.4069

CH4_B     1         1         RESOURCE FOR CHANNEL CONNECTOR
          ATTR.     INITIAL
          NAME      VALUE
          ========  ========
          RATE      0.4069

CH5_A     1         1         RESOURCE FOR CHANNEL CONNECTOR
          ATTR.     INITIAL
          NAME      VALUE
          ========  ========
          RATE      0.4069

CH5_B     1         1         RESOURCE FOR CHANNEL CONNECTOR
          ATTR.     INITIAL
          NAME      VALUE
          ========  ========
          RATE      0.4069

CH6_A     1         1         RESOURCE FOR CHANNEL CONNECTOR
          ATTR.     INITIAL
          NAME      VALUE
          ========  ========
          RATE      VRATE

CH6_B     1         1         RESOURCE FOR CHANNEL CONNECTOR
          ATTR.     INITIAL
          NAME      VALUE
```

```
PAGE     4
          ======== ========
          RATE     VRATE

CH7_A     1        1         RESOURCE FOR CHANNEL CONNECTOR
          ATTR.    INITIAL
          NAME     VALUE
          ======== ========
          RATE     VRATE

CH7_B     1        1         RESOURCE FOR CHANNEL CONNECTOR
          ATTR.    INITIAL
          NAME     VALUE
          ======== ========
          RATE     VRATE

CH8_A     1        1         RESOURCE FOR CHANNEL CONNECTOR
          ATTR.    INITIAL
          NAME     VALUE
          ======== ========
          RATE     VRATE

CH8_B     1        1         RESOURCE FOR CHANNEL CONNECTOR
          ATTR.    INITIAL
          NAME     VALUE
          ======== ========
          RATE     VRATE

CH9_A     1        1         RESOURCE FOR CHANNEL OCCONNECTOR
          ATTR.    INITIAL
          NAME     VALUE
          ======== ========
          RATE     VRATE

CH9_B     1        1         RESOURCE FOR CHANNEL CONNECTOR
          ATTR.    INITIAL
          NAME     VALUE
          ======== ========
          RATE     VRATE

CHQ       1        1         COMMAND HEAD-QUARTERS
          ATTR.    INITIAL
          NAME     VALUE
          ======== ========
          M_ROUTE  8

DK1       1        1         DISK FOR COMMAND HEAD-QUARTERS
          ATTR.    INITIAL
          NAME     VALUE
          ======== ========
```

```
PAGE     5
         LATDELTA 15
         LATENCY  15
         M_ROUTE  0
         SEEK     TIME1
         SPEED    20000

HQ       1        1        HEAD-QUARTERS
         ATTR.    INITIAL
         NAME     VALUE
         ======== ========
         M_ROUTE  8

L3       1        1        RESOURCE FOR NODE
         ATTR.    INITIAL
         NAME     VALUE
         ======== ========
         M_ROUTE  8

SW1      1        1        SWITCH BETWEEN AIRBASES AND OTHER TWO SWITCHES (1&2)
         ATTR.    INITIAL
         NAME     VALUE
         ======== ========
         M_ROUTE  8

SW2      1        1        SWITCH BETWEEN SWITCH 1 & 3 AND HQ
         ATTR.    INITIAL
         NAME     VALUE
         ======== ========
         M_ROUTE  8

SW3      1        1        SWITCH BETWEEN SWITCH 1 & 2 AND CHQ
         ATTR.    INITIAL
         NAME     VALUE
         ======== ========
         M_ROUTE  8

ARCHITECTURE LEGAL PATH DEFINITION

FROM      TO        NEXT      VIA
DEVICE    DEVICE    DEVICE    LINK
========  ========  ========  ========
AB1       AB2       SW1       CH1_A
AB1       CHQ       SW1       CH1_A
AB1       DK1       SW1       CH1_A
AB1       HQ        SW1       CH1_A
AB1       L3        SW1       CH1_A
AB1       SW1       SW1       CH1_A
AB1       SW2       SW1       CH1_A
AB1       SW3       SW1       CH1_A
```

AB2	AB1	SW1	CH2_A
AB2	CHQ	SW1	CH2_A
AB2	DK1	SW1	CH2_A
AB2	HQ	SW1	CH2_A
AB2	L3	SW1	CH2_A
AB2	SW1	SW1	CH2_A
AB2	SW2	SW1	CH2_A
AB2	SW3	SW1	CH2_A
CHQ	AB1	SW3	CH8_A
CHQ	AB2	SW3	CH8_A
CHQ	DK1	DK1	CH9_A
CHQ	HQ	SW3	CH8_A
CHQ	L3	SW3	CH8_A
CHQ	SW1	SW3	CH8_A
CHQ	SW2	SW3	CH8_A
CHQ	SW3	SW3	CH8_A
DK1	AB1	CHQ	CH9_B
DK1	AB2	CHQ	CH9_B
DK1	CHQ	CHQ	CH9_B
DK1	HQ	CHQ	CH9_B
DK1	L3	CHQ	CH9_B
DK1	SW1	CHQ	CH9_B
DK1	SW2	CHQ	CH9_B
DK1	SW3	CHQ	CH9_B
HQ	AB1	SW2	CH7_A
HQ	AB2	SW2	CH7_A
HQ	CHQ	SW2	CH7_A
HQ	DK1	SW2	CH7_A
HQ	L3	L3	CH6_A
HQ	SW1	SW2	CH7_A
HQ	SW2	SW2	CH7_A
HQ	SW3	SW2	CH7_A
L3	AB1	HQ	CH6_B
L3	AB2	HQ	CH6_B
L3	CHQ	HQ	CH6_B
L3	DK1	HQ	CH6_B
L3	HQ	HQ	CH6_B
L3	SW1	HQ	CH6_B
L3	SW2	HQ	CH6_B
L3	SW3	HQ	CH6_B
SW1	AB1	AB1	CH1_B
SW1	AB2	AB2	CH2_B
SW1	CHQ	SW3	CH4_A
SW1	DK1	SW3	CH4_A
SW1	HQ	SW2	CH3_A
SW1	L3	SW2	CH3_A
SW1	SW2	SW2	CH3_A
SW1	SW3	SW3	CH4_A
SW2	AB1	SW1	CH3_B

```
SW2      AB2      SW1      CH3_B
SW2      CHQ      SW3      CH5_A
SW2      DK1      SW3      CH5_A
SW2      HQ       HQ       CH7_B
SW2      L3       HQ       CH7_B
SW2      SW1      SW1      CH3_B
SW2      SW3      SW3      CH5_A
SW3      AB1      SW1      CH4_B
SW3      AB2      SW1      CH4_B
SW3      CHQ      CHQ      CH8_B
SW3      DK1      CHQ      CH8_B
SW3      HQ       SW2      CH5_B
SW3      L3       SW2      CH5_B
SW3      SW1      SW1      CH4_B
SW3      SW2      SW2      CH5_B

ACTION DEFINITION.....

ACTION
MNEMONIC          COMMENT
======== ======== ==================================================
FORMAT            TTIME USED TO FORMAT PLANS FROM CHQ
LATENCY           LATENCY PAUSE SUBSEQUENT TO SEEK
ROUTE_OH          PROCESSING DELAY TO ROUTE A MESSAGE
SEEK              SEEKING INFORMATION ON DISK
UPDATE            UPDATING INFO SINCE PREVIOUS BROADCAST TO OTHER NODES
XFER              TRANSFER INFORMATION SOUGHT ON DISK
XFER_OH           PROCESSING DELAY TO ROUTE A MESSAGE OVER A CHANNEL

PROCESS DEFINITION.....

PROCESS
MNEMONIC          DESCRIPTION
======== ======== ==================================================
ABUPDATE          UPDATE DATA FROM AIRBASE


ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
======== ======== ======== ======== ======== =========================
         START             NO
         GIVEN    MSG
         RETURN   MSG
         UPDATE   CONSTANT Ø.1               TIME CONSUMED IN UPDATING
                  US       RESUME
         END

LOCAL VARIABLES OF PROCESS ABUPDATE
==================================================================
    1 MSG       (I)     2 UPDATE     (A)
```

```
PAGE     8
PROCESS
MNEMONIC           DESCRIPTION
======== ======== ==================================================
AB_DATA            AIR BASE STATUS BROUADCAST TO ALL OTHER NODES


ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
======== ======== ======== ======== ======== =======================
         START             NO
         GIVEN    MSG
         RETURN   MSG
         CALL     MRS      NOWAIT   10       PROCESS REQUEST TO CHQ
         GIVEN    CHQ_DATA 10       $REQNORE
                  750      CHQ      MSG
         CALL     MRS      NOWAIT   10       PROCESS REQUEST TO HQ
         GIVEN    HQ_DATA  10       $REQNORE
                  750      HQ       MSG
         ASSIGN   $CNODE                     CURRENT NODE
                  CNODE
         COMPARE  CNODE             EQ       TEST FOR CURRENT NODE
                  AB1               AB1
         CALL     MRS      NOWAIT   10       PROCESS REQUEST TO AB1
         GIVEN    ABUPDATE 10       $REQNORE
                  750      AB1      MSG
         BRANCH   END      100               BRANCH TO THE END
AB1      ENTRY                               ENTRY FROM COMPARE NODE
         CALL     MRS      NOWAIT   10       PROCESS REQUEST TO AB2
         GIVEN    ABUPDATE 10       $REQNORE
                  750      AB2      MSG
END      ENTRY                               ENDTRY FROM REQUEST TO A1H
         END

LOCAL VARIABLES OF PROCESS AB_DATA
====================================================================
    1 MSG       (I)     2 MRS         (P)     3 CHQ_DATA  (P)     4 CHQ         (R)
    5 HQ_DATA   (P)     6 HQ          (R)     7 CNODE             8 AB1         (R)
    9 ABUPDATE  (P)    10 AB2         (R)
PROCESS
MNEMONIC           DESCRIPTION
======== ======== ==================================================
AB_REQ             AIRBASE REQUEST FOR PLANS REPORT FROM CHQ


ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
======== ======== ======== ======== ======== =======================
         START             NO
         GIVEN    MSG
         RETURN   MSG
         CALL     MRS      WAIT     5        PROCESS REQUEST TO CHQ
```

```
PAGE    9
         GIVEN    PLANS    5        $REQRESP
                  200      CHQ      MSG
         END

LOCAL VARIABLES OF PROCESS AB_REQ
=================================================================
    1 MSG       (I)     2 MRS        (P)     3 PLANS      (P)     4 CHQ        (R)

PROCESS
MNEMONIC          DESCRIPTION
======== ======== =================================================
CHANPROC          FULL AND HALF DUPLEX CHANNEL LOGIC

ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
======== ======== ======== ======== ======== ======================
         START    ALL      NO
         GIVEN    MSG
         ASSIGN   MSG      CNODE             SET INTERNAL NODE CURRENT
                  $CNODE
         ASSIGN   MSG      TNODE             GET DESTINATION NODE
                  TO_NODE
         ASSIGN   $NXTNODE TO_NODE           SET NEXT NODE TO DESTN
                  NXT_NODE
         ASSIGN   $CHANNEL TO_NODE           GET CHANNEL TO NEXT NODE
                  CHANNEL
         ALLOC    CHANNEL  1        ALL      OBTAIN CHANNEL FOR XFER
                  $PRIORTY
         ASSIGN   CHANNEL  RATE              WHAT IS CHANNEL RATE?
                  VSPEED
         ASSIGN   MSG      LENGTH            MESSAGE LENGTH
                  VLENGTH
         EVAL     VM_OVHD                    CALCULATE TRANSFER TIME
                  VSPEED*VLENGTH
         XFER_OH  CONSTANT VM_OVHD           DELAY DUE TO TRANSFER TIME
                  US       RESUME
         ASSIGN   NXT_NODE                   MSG RESIDES IN NEXT NODE
                  MSG      CNODE
         ASSIGN   NXT_NODE                   SET INTERNAL NODE REGISTER
                  $CNODE
         DEALLOC  CHANNEL  1                 FREE UP CHANNEL AFTER XFER
         CALL     NODEPROC WAIT     0        ROUTE MESSAGE TO NEXT NODE
         GIVEN    MSG
         END

LOCAL VARIABLES OF PROCESS CHANPROC
=================================================================
    1 MSG       (I)     2 TO_NODE           3 NXT_NODE          4 CHANNEL
    5 VSPEED            6 VLENGTH           7 VM_OVHD           8 XFER_OH    (A)
```

```
    9 NODEPROC   (P)
PROCESS
MNEMONIC          DESCRIPTION
======== ======== ==================================================
CHQ_DATA          CHQ GETS MESSAGE, FORMULATES RESPONSE, AND REPLIES

ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
======== ======== ======== ======== ======== ========================
         START             NO
         GIVEN    MSG
         RETURN   MSG
         ASSIGN   MSG      LENGTH            MAKE MSG-LENGTH = V_LENGTH
                  V_LENGTH
         EVAL     V_TIME                     EVALUATE MSG PROCESS TIME
                  .015*V_LENGTH
         UPDATE   CONSTANT V_TIME            PROCESSING TIME CONSUMED
                  US       RESUME
         END

LOCAL VARIABLES OF PROCESS CHQ_DATA
==================================================================
    1 MSG        (I)    2 V_LENGTH        3 V_TIME          4 UPDATE     (A)

PROCESS
MNEMONIC          DESCRIPTION
======== ======== ==================================================
DESTPROC          PROCESSING AT DESTINATION OF MESSAGE

ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
======== ======== ======== ======== ======== ========================
         START    ALL      NO
         GIVEN    MSG
         ASSIGN   MSG      CNODE             CURRENT NODE
                  C_NODE
         COMPARE  MSG      TYPE     EQ       IF RESPONSE,D DESTROY
                  $RESP             DESTROY
         ALLOC    C_NODE   1        ALL      ALLOCATE CURRENT NODE
                  $PRIORTY
         ASSIGN   MSG      RPROC             EXECUTE THE CALLED PROCESS
                  PROCESS
         ASSIGN   MSG      RPROCPRI          SET PRIORITY FOR REQ PROC
                  PRIORITY
         CALL     PROCESS  WAIT     PRIORITY WAIT UNTIL COMPLETE
         GIVEN    MSG
         RETURN   MSG
         DEALLOC  C_NODE   1                 DEALLOCATE CURRENT NODE
         COMPARE  MSG      TYPE     EQ       NO RESPONSE REQ -> DESTROY
```

```
                  $REQNORE           DESTROY
         ASSIGN   $RESP                       CHANGE MSG RESPONSE TYPE
                  MSG       TYPE
         ASSIGN   MSG       FNODE             SWITCH FROM AND TO NODES
                  MSG       TNODE
         ASSIGN   MSG       CNODE             CURRENT NODE IS FROM NODE
                  MSG       FNODE
         CALL     CHANPROC  WAIT     0        RETURN MESSAGE TO ORIGIN
         GIVEN    MSG
         BRANCH   END       100
DESTROY  ENTRY                                TERMINATE MESSAGE AT DEST
         DESTROY  MSG                         TERMINATE MSG
END      ENTRY
         END

LOCAL VARIABLES OF PROCESS DESTPROC
===================================================================
    1 MSG       (I)    2 C_NODE          3 PROCESS    (X)    4 PRIORITY
    5 CHANPROC  (P)
PROCESS
MNEMONIC          DESCRIPTION
======== ======== ==================================================
DISK_OP           OPERATION OF DISK

ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
======== ======== ======== ======== ======== =========================
         START             NO
         GIVEN    LENGTH   DISK
         ASSIGN   DISK     SPEED             MAKE DISK SPEED = V_SPEED
                  V_SPEED
         EVAL     XFERTIME                   TRANSFER TIME CALCULATED
                  LENGTH/V_SPEED
         ALLOC    DISK     1        PARTIAL  DISK ALLOCATED
                  $PRIOPTY
         ASSIGN   DISK     SEEK              MAKE SEEKTIME = SEEK
                  SEEKTIME
         SEEK     UNIFORM  SEEKTIME SEEKTIME TIME FOR SEEK IS CONSUMED
                  US       RESUME
         ASSIGN   DISK     LATENCY           MAKE DISK LATENCY=LATETIME
                  LATETIME
         LATENCY  UNIFORM  LATETIME LATETIME TIME CONSUMED FOR LATENCY
                  US       RESUME
         XFER     CONSTANT XFERTIME          TRANSFER TIME CONSUMED
                  US       RESUME
         DEALLOC  DISK     1                 DISK RESOURCE DEALLOCATED
         END

LOCAL VARIABLES OF PROCESS DISK_OP
```

```
==================================================================
    1 LENGTH           2 DISK              3 V_SPEED           4 XFERTIME
    5 SEEKTIME         6 SEEK       (A)    7 LATETIME          8 LATENCY     (A)
    9 XFER        (A)
PROCESS
MNEMONIC          DESCRIPTION
======== ======== ==================================================
HQ_DATA           HQ GETS MESSAGE, FORMULATES RESPONSE, AND REPLIES


ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
======== ======== ======== ======== ======== =========================
         START             NO
         GIVEN    MSG
         RETURN   MSG
         ASSIGN   MSG      LENGTH            MAKE MSG_LENGTH = V_LENGTH
                  V_LENGTH
         EVAL     V_TIME                     EVALUATE MSG PROCESS TIME
                  .016*V_LENGTH
         UPDATE   CONSTANT V_TIME            PROCESSING TIME CONSUMED
                  US       RESUME
         END

LOCAL VARIABLES OF PROCESS HQ_DATA
==================================================================
    1 MSG         (I)    2 V_LENGTH          3 V_TIME            4 UPDATE      (A)

PROCESS
MNEMONIC          DESCRIPTION
======== ======== ==================================================
HQ_REQ            HQ REQUEST FOR STATUS DISPLAY FROM CHQ


ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
======== ======== ======== ======== ======== =========================
         START    L3       NO
         GIVEN    MSG
         RETURN   MSG
         CALL     MRS      WAIT     4        MAKES I/O REQUEST TO CHQ
         GIVEN    PLANS    4        $REQRESP
                  200      CHQ      MSG
         END

LOCAL VARIABLES OF PROCESS HQ_REQ
==================================================================
    1 MSG         (I)    2 MRS        (P)    3 PLANS      (P)    4 CHQ         (R)

PROCESS
MNEMONIC          DESCRIPTION
```

```
======== ======== ==================================================
MRS               GENERATE A PROCESS REQUEST MESSAGE AND INITIATE I/O

ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
======== ======== ======== ======== ======== ========================
         START    ALL      NO
         GIVEN    PROCESS  PRIORITY MSG_TYPE
                  MSG_LNTH TO_NODE  MSG
         CREATE   MSG                        CREATE MESSAGE
         ASSIGN   MSG_LNTH                   SET MESSAGE LENGTH
                  MSG      LENGTH
         ASSIGN   PROCESS                    SET PROCESS
                  MSG      RPROC
         ASSIGN   PRIORITY                   SET PRIORITY
                  MSG      RPROCPRI
         ASSIGN   TO_NODE                    SET DESTINATION
                  MSG      TNODE
         ASSIGN   MSG_TYPE                   SET MESSAGE TYPE
                  MSG      TYPE
         CALL     NODEPROC WAIT     0        EXECUTIVE SERVICING OF MSG
         GIVEN    MSG
         END

LOCAL VARIABLES OF PROCESS MRS
=====================================================================
    1 PROCESS    (X)    2 PRIORITY          3 MSG_TYPE          4 MSG_LNTH
    5 TO_NODE           6 MSG       (I)     7 NODEPROC   (P)
PROCESS
MNEMONIC          DESCRIPTION
======== ======== ==================================================
NODEPROC          NODAL PROCESSING AND ROUTING

ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
======== ======== ======== ======== ======== ========================
         START    ALL      NO
         GIVEN    MSG
         ASSIGN   MSG      CNODE             INDICATE CURRENT NODE
                  C_NODE
         ASSIGN   C_NODE   M_ROUTE           PROCESSING RATE OF NODE
                  RT_OVHD
         ASSIGN   MSG      LENGTH            GET MESSAGE LENGTH
                  MSG_LNTH
         EVAL     OVERHEAD                   COMPUTE PROCESSING DELAY
                  MSG_LNTH*RT_OVHD
         ALLOC    C_NODE   1        ALL      ALLOCATE CURRENT NODE
                  $PRIORTY
ROUTE_OH CONSTANT OVERHEAD                   DELAY FOR ROUTING
```

```
                  US        RESUME
         DEALLOC  C_NODE    1                  RELEASE C_NODE TO OTHERS
         COMPARE  MSG       CNODE     EQ       IS MSG AT DESTINATION?
                  MSG       TNODE     CONTROL
         CALL     CHANPROC  WAIT      0        FORWARD MSG TO CHANNEL
         GIVEN    MSG
         BRANCH   END       100
CONTROL  ENTRY                                 MESSAGE AT DESTINATION
         CALL     DESTPROC  WAIT      0        CONTEXT SWITCH MESSAGE
         GIVEN    MSG
END      ENTRY
         END

LOCAL VARIABLES OF PROCESS NODEPROC
==================================================================
    1 MSG         (I)    2 C_NODE             3 RT_OVHD            4 MSG_LNTH
    5 OVERHEAD           6 ROUTE_OH   (A)     7 CHANPROC   (P)     8 DESTPROC   (P)

PROCESS
MNEMONIC          DESCRIPTION
======== ======== ================================================
PLANS             REQUEST FOR PLANS FROM CHQ

ENTRY    OPCODE   PARM      PARM      PARM     COMMENT
======== ======== ======== ======== ======== ======================
         START    CHQ       NO
         GIVEN    MSG
         RETURN   MSG
         ASSIGN   MSG       LENGTH             MAKE MSG LENGTH = V_LENGTH
                  V_LENGTH
         EVAL     V_TIME                       EVALUATE MSG PROCESS TIME
                  .01*V_LENGTH
         FORMAT   CONSTANT  V_TIME             TIME USED TO FORMAT PLANS
                  US        RESUME
         CALL     DISK_OP   WAIT      10       CALLING PROCESS DISK_OP
         GIVEN    500       DK1
         ASSIGN   500                          INCREASE MSG LENGTH
                  MSG       LENGTH
         END

LOCAL VARIABLES OF PROCESS PLANS
==================================================================
    1 MSG         (I)    2 V_LENGTH           3 V_TIME             4 FORMAT     (A)
    5 DISK_OP     (P)    6 DK1        (R)
PROCESS
MNEMONIC          DESCRIPTION
======== ======== ================================================
TRACE             TURN ON TRACE OUTPUT
```

```
ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
======== ======== ======== ======== ======== =======================
         START    ALL      NO
         COMPARE  V_TRACE           EQ       TEST IF FLAG SET FOR TRACE
                  Ø                 NOTRACE
         TRACE    ON
NOTRACE  ENTRY
         END

GLOBAL VARIABLES OF PROCESS TRACE
=======================================================================
    1 V_TRACE    (C)

LOAD DEFINITION.....

LOAD
MNEMONIC          DESCRIPTION
======== ======== =====================================================
ABLOAD            COMMUNICATIONS FROM AIRBASES
         LOAD     NODES
         ======== ======== ======== ========
         AB1      AB2


         PROCESS           SCHEDULE
         MNEMONIC MAX #    METHOD   MEAN     DELTA    UNITS    PRIORITY
         ======== ======== ======== ======== ======== ======== ========
         AB_DATA  6Ø       INTERVAL ABDRATE           US       1Ø
         AB_REQ   6Ø       EXPONENT ABRRATE           US       5
LOAD
MNEMONIC          DESCRIPTION
======== ======== =====================================================
HQLOAD            REQUEST DATA FROM CHQ
         LOAD     NODES
         ======== ======== ======== ========
         L3


         PROCESS           SCHEDULE
         MNEMONIC MAX #    METHOD   MEAN     DELTA    UNITS    PRIORITY
         ======== ======== ======== ======== ======== ======== ========
         HQ_REQ   6Ø       EXPONENT HQRRATE           US       4

SCENARIO DEFINITION....

SCENARIO
MNEMONIC          DESCRIPTION
```

```
PAGE   10
======== ======== ==================================================
TEST01            SCENARIO FOR MINI MITRE 1

PERIOD   PERIOD   OUTPUT
LENGTH   UNITS    UNITS
======== ======== ========
360000   US       US

         PERIOD   PERIOD   PERIOD   PERIOD   PERIOD   PERIOD   PERIOD
         MNEMONIC MNEMONIC MNEMONIC MNEMONIC MNEMONIC MNEMONIC MNEMONIC
======== ======== ======== ======== ======== ======== ======== ========
         1        2        3        4        5        6        7
         PERIOD   PERIOD   PERIOD   PERIOD   PERIOD   PERIOD   PERIOD
         MNEMONIC MNEMONIC MNEMONIC MNEMONIC MNEMONIC MNEMONIC MNEMONIC
======== ======== ======== ======== ======== ======== ======== ========
         8        9        10

TRIGGER  TIME  TO SCHEDULE SCHEDULE TRIGGER  TIME  TO SCHEDULE SCHEDULE
MNEMONIC SCHEDULE UNITS    PRIORITY MNEMONIC SCHEDULE UNITS    PRIORITY
======== ======== ======== ======== ======== ======== ======== ========
ABLOAD   0        US       0        HQLOAD   0        US       0
TRACE    0        US       0

####    0 ERRORS WERE DETECTED DURING MODEL INITIALIZATION
```

```
SIMULATION TIME =     3600000.00000 US

CONSTANT REPORT

         CURRENT
CONSTANT VALUE...
======== ==========
V_TRACE        0.
```

PAGE 19

SIMULATION TIME = 3600000.00000 US

VARIABLE REPORT

NUMERIC VARIABLES...

VARIABLE	TOTAL SAMPLES.	VALUE CURRENT...	MEAN......	STD DEV...	MINIMUM...	MAXIMUM...
ABDRATE	1	60000.000	60000.000	0.	60000.000	60000.000
ABRRATE	1	36000.000	36000.000	0.	36000.000	36000.000
HQRRATE	1	72000.000	72000.000	0.	72000.000	72000.000
TIME1	1	30.000	30.000	0.	30.000	30.000
VRATE	1	1.628	1.628	0.	1.628	1.628

NON-NUMERIC VARIABLES...

VARIABLE	CURRENT TYPE	CURRENT VALUE

PAGE 20

SIMULATION TIME = 3600000.00000 US

ITEM REPORT

ITEM NAME	NUMBER CREATED	NUMBER DESTR'D	TIME IN SYSTEM MINIMUM...	MAXIMUM...	AVERAGE...	STD DEV...
MSG	522	522	22606.59	194855.44	71890.94	38971.35

```
PAGE   21

SIMULATION TIME =     3600000.00000 US

RESOURCE REPORT

                  TOTAL
RESOURCE          NUMBER   CURRENT... MEAN...... STD DEV... MINIMUM... MAXIMUM...
=============== ======== ========== ========== ========== ========== ==========
AB1
         # IDLE               1.000      0.513      0.500      0.         1.000
     REQUEST TIME                     4694.304   5663.327      0.     22000.008
        HOLD TIME        0               0.         0.         0.         0.

        INTO BUSY      415
      OUT OF BUSY      415
           # BUSY                0.      0.487      0.500      0.         1.000
        BUSY TIME                     4221.696   2309.316      0.      6000.008

       # INACTIVE                0.         0.         0.         0.         0.

        INTO WAIT      415
      OUT OF WAIT      415
        # WAITING                0.      0.541      0.973      0.         5.000
        WAIT TIME                     4694.304   5663.327      0.     22000.008

      CURRENTLY ALLOCATED
             TO PROCESSES: NONE

      PROCESSES CURRENTLY
                  WAITING: NONE

                  TOTAL
RESOURCE          NUMBER   CURRENT... MEAN...... STD DEV... MINIMUM... MAXIMUM...
=============== ======== ========== ========== ========== ========== ==========
AB2
         # IDLE               1.000      0.513      0.500      0.         1.000
     REQUEST TIME                     4329.880   5316.789      0.     22000.000
        HOLD TIME        0               0.         0.         0.         0.

        INTO BUSY      415
      OUT OF BUSY      415
           # BUSY                0.      0.487      0.500      0.         1.000
        BUSY TIME                     4221.696   2309.317      0.      6000.000

       # INACTIVE                0.         0.         0.         0.         0.

        INTO WAIT      415
      OUT OF WAIT      415
        # WAITING                0.      0.499      0.911      0.         5.000
```

```
PAGE   22
    WAIT TIME                              4329.880   5316.789      0.       22000.000

    CURRENTLY ALLOCATED
           TO PROCESSES: NONE

    PROCESSES CURRENTLY
                WAITING: NONE

                TOTAL
RESOURCE        NUMBER   CURRENT... MEAN...... STD DEV... MINIMUM... MAXIMUM...
=============== ======== ========== ========== ========== ========== ==========
CH1_A
         # IDLE               1.000      0.935      0.247      0.         1.000
   REQUEST TIME                          0.         0.         0.         0.
      HOLD TIME         0                0.         0.         0.         0.

      INTO BUSY       237
    OUT OF BUSY       237
         # BUSY               0.         0.065      0.247      0.         1.000
      BUSY TIME                        994.094    389.263    325.500   1220.750

     # INACTIVE               0.         0.         0.         0.         0.

      INTO WAIT       237
    OUT OF WAIT       237
      # WAITING               0.         0.         0.         0.         1.000
      WAIT TIME                          0.         0.         0.         0.

    CURRENTLY ALLOCATED
           TO PROCESSES: NONE

    PROCESSES CURRENTLY
                WAITING: NONE

                TOTAL
RESOURCE        NUMBER   CURRENT... MEAN...... STD DEV... MINIMUM... MAXIMUM...
=============== ======== ========== ========== ========== ========== ==========
CH1_B
         # IDLE               1.000      0.966      0.180      0.         1.000
   REQUEST TIME                          0.         0.         0.         0.
      HOLD TIME         0                0.         0.         0.         0.

      INTO BUSY       119
    OUT OF BUSY       119
         # BUSY               0.         0.034      0.180      0.         1.000
      BUSY TIME                       1015.532    203.448    813.750   1220.750

     # INACTIVE               0.         0.         0.         0.         0.
```

```
PAGE   23
       INTO WAIT       119
     OUT OF WAIT       119
       # WAITING                 0.         0.         0.         0.          1.000
       WAIT TIME                            0.         0.         0.          0.

     CURRENTLY ALLOCATED
            TO PROCESSES: NONE

     PROCESSES CURRENTLY
                WAITING: NONE

                       TOTAL
RESOURCE               NUMBER   CURRENT... MEAN...... STD DEV... MINIMUM... MAXIMUM...
============== ======== ========== ========== ========== ========== ==========
CH2_A
          # IDLE                 1.000      0.935      0.247      0.         1.000
    REQUEST TIME                            0.         0.         0.         0.
       HOLD TIME          0                 0.         0.         0.         0.

       INTO BUSY        237
     OUT OF BUSY        237
          # BUSY                 0.         0.065      0.247      0.         1.000
       BUSY TIME                          994.092    389.262    325.500   1220.750

      # INACTIVE                 0.         0.         0.         0.         0.

       INTO WAIT        237
     OUT OF WAIT        237
       # WAITING                 0.         0.         0.         0.         1.000
       WAIT TIME                            0.         0.         0.         0.

     CURRENTLY ALLOCATED
            TO PROCESSES: NONE

     PROCESSES CURRENTLY
                WAITING: NONE

                       TOTAL
RESOURCE               NUMBER   CURRENT... MEAN...... STD DEV... MINIMUM... MAXIMUM...
============== ======== ========== ========== ========== ========== ==========
CH2_B
          # IDLE                 1.000      0.966      0.180      0.         1.000
    REQUEST TIME                            0.         0.         0.         0.
       HOLD TIME          0                 0.         0.         0.         0.

       INTO BUSY        119
     OUT OF BUSY        119
          # BUSY                 0.         0.034      0.180      0.         1.000
       BUSY TIME                         1016.533    203.447    813.750   1220.750
```

```
      # INACTIVE                    0.        0.        0.        0.        0.

      INTO WAIT       119
    OUT OF WAIT       119
      # WAITING                     0.        0.        0.        0.        1.000
      WAIT TIME                               0.        0.        0.        0.

    CURRENTLY ALLOCATED
            TO PROCESSES: NONE

    PROCESSES CURRENTLY
                WAITING: NONE

                      TOTAL
RESOURCE              NUMBER    CURRENT... MEAN...... STD DEV... MINIMUM... MAXIMUM...
===============       ========  ========== ========== ========== ========== ==========
CH3_A
            # IDLE                1.000     0.990     0.100     0.        1.000
     REQUEST TIME                           0.        0.        0.        0.
        HOLD TIME         0                 0.        0.        0.        0.

      INTO BUSY       118
    OUT OF BUSY       118
          # BUSY                  0.        0.010     0.100     0.        1.000
        BUSY TIME                         305.088     0.079   305.000   305.188

      # INACTIVE                  0.        0.        0.        0.        0.

      INTO WAIT       118
    OUT OF WAIT       118
      # WAITING                   0.        0.        0.        0.        1.000
      WAIT TIME                             0.        0.        0.        0.

    CURRENTLY ALLOCATED
            TO PROCESSES: NONE

    PROCESSES CURRENTLY
                WAITING: NONE

                      TOTAL
RESOURCE              NUMBER    CURRENT... MEAN...... STD DEV... MINIMUM... MAXIMUM...
===============       ========  ========== ========== ========== ========== ==========
CH3_B
            # IDLE                1.000     1.000     0.        1.000     1.000
     REQUEST TIME                           0.        0.        0.        0.
        HOLD TIME         0                 0.        0.        0.        0.

      INTO BUSY           0
```

```
    OUT OF BUSY          0
          # BUSY                  0.        0.         0.         0.         0.
       BUSY TIME                            0.         0.         0.         0.

      # INACTIVE                  0.        0.         0.         0.         0.

       INTO WAIT         0
     OUT OF WAIT         0
       # WAITING                  0.        0.         0.         0.         0.
       WAIT TIME                            0.         0.         0.         0.

    CURRENTLY ALLOCATED
           TO PROCESSES: NONE

    PROCESSES CURRENTLY
                WAITING: NONE

                    TOTAL
RESOURCE            NUMBER   CURRENT... MEAN...... STD DEV... MINIMUM... MAXIMUM...
=============== ======== ========== ========== ========== ========== ==========
CH4_A
          # IDLE                1.000      0.987      0.112      0.         1.000
    REQUEST TIME                           0.         0.         0.         0.
       HOLD TIME         0                 0.         0.         0.         0.

       INTO BUSY       238
     OUT OF BUSY       238
          # BUSY                0.         0.013      0.112      0.         1.000
       BUSY TIME                         192.294    111.850     81.376    305.188

      # INACTIVE                0.         0.         0.         0.         0.

       INTO WAIT       238
     OUT OF WAIT       238
       # WAITING                0.         0.         0.         0.         1.000
       WAIT TIME                           0.         0.         0.         0.

    CURRENTLY ALLOCATED
           TO PROCESSES: NONE

    PROCESSES CURRENTLY
                WAITING: NONE

                    TOTAL
RESOURCE            NUMBER   CURRENT... MEAN...... STD DEV... MINIMUM... MAXIMUM...
=============== ======== ========== ========== ========== ========== ==========
CH4_B
          # IDLE                1.000      0.993      0.082      0.         1.000
    REQUEST TIME                           0.         0.         0.         0.
```

```
     HOLD TIME          0                   0.         0.         0.         0.

     INTO BUSY        120
   OUT OF BUSY        120
        # BUSY                 0.           0.007      0.082      0.         1.000
     BUSY TIME                            203.448      0.037    203.375    203.500

    # INACTIVE                 0.           0.         0.         0.         0.

     INTO WAIT        120
   OUT OF WAIT        120
     # WAITING                 0.           0.         0.         0.         1.000
     WAIT TIME                              0.         0.         0.         0.

   CURRENTLY ALLOCATED
          TO PROCESSES: NONE

   PROCESSES CURRENTLY
              WAITING: NONE

                TOTAL
RESOURCE        NUMBER   CURRENT... MEAN...... STD DEV... MINIMUM... MAXIMUM...
=============== ======== ========== ========== ========== ========== ==========
CH5_A
        # IDLE                 1.000        0.999      0.033      0.         1.000
  REQUEST TIME                              0.         0.         0.         0.
     HOLD TIME          0                   0.         0.         0.         0.

     INTO BUSY         48
   OUT OF BUSY         48
        # BUSY                 0.           0.001      0.033      0.         1.000
     BUSY TIME                             81.408      0.078     81.250     81.500

    # INACTIVE                 0.           0.         0.         0.         0.

     INTO WAIT         48
   OUT OF WAIT         48
     # WAITING                 0.           0.         0.         0.         1.000
     WAIT TIME                              0.         0.         0.         0.

   CURRENTLY ALLOCATED
          TO PROCESSES: NONE

   PROCESSES CURRENTLY
              WAITING: NONE

                TOTAL
RESOURCE        NUMBER   CURRENT... MEAN...... STD DEV... MINIMUM... MAXIMUM...
=============== ======== ========== ========== ========== ========== ==========
```

```
CH5_B
        # IDLE                   1.000      0.997      0.052      0.         1.000
     REQUEST TIME                           0.         0.         0.         0.
        HOLD TIME          0                0.         0.         0.         0.

        INTO BUSY         48
      OUT OF BUSY         48
           # BUSY                0.         0.003      0.052      0.         1.000
        BUSY TIME                         203.447      0.083    203.250    203.500

       # INACTIVE                0.         0.         0.         0.         0.

        INTO WAIT         48
      OUT OF WAIT         48
        # WAITING                0.         0.         0.         0.         1.000
        WAIT TIME                           0.         0.         0.         0.

      CURRENTLY ALLOCATED
             TO PROCESSES: NONE

      PROCESSES CURRENTLY
                  WAITING: NONE

                 TOTAL
RESOURCE         NUMBER   CURRENT... MEAN...... STD DEV... MINIMUM... MAXIMUM...
================ ======== ========== ========== ========== ========== ==========
CH6_A
           # IDLE                1.000      0.989      0.104      0.         1.000
     REQUEST TIME                           0.         0.         0.         0.
        HOLD TIME          0                0.         0.         0.         0.

        INTO BUSY         48
      OUT OF BUSY         48
           # BUSY                0.         0.011      0.104      0.         1.000
        BUSY TIME                         813.798      0.076    813.750    814.000

       # INACTIVE                0.         0.         0.         0.         0.

        INTO WAIT         48
      OUT OF WAIT         48
        # WAITING                0.         0.         0.         0.         1.000
        WAIT TIME                           0.         0.         0.         0.

      CURRENTLY ALLOCATED
             TO PROCESSES: NONE

      PROCESSES CURRENTLY
                  WAITING: NONE
```

RESOURCE	TOTAL NUMBER	CURRENT...	MEAN......	STD DEV...	MINIMUM...	MAXIMUM...
CH6_B						
# IDLE		1.000	0.996	0.066	0.	1.000
REQUEST TIME			0.	0.	0.	0.
HOLD TIME	0		0.	0.	0.	0.
INTO BUSY	48					
OUT OF BUSY	48					
# BUSY		0.	0.004	0.066	0.	1.000
BUSY TIME			325.524	0.064	325.500	325.750
# INACTIVE		0.	0.	0.	0.	0.
INTO WAIT	48					
OUT OF WAIT	48					
# WAITING		0.	0.	0.	0.	1.000
WAIT TIME			0.	0.	0.	0.

CURRENTLY ALLOCATED
TO PROCESSES: NONE

PROCESSES CURRENTLY
WAITING: NONE

RESOURCE	TOTAL NUMBER	CURRENT...	MEAN......	STD DEV...	MINIMUM...	MAXIMUM...
CH7_A						
# IDLE		1.000	0.996	0.066	0.	1.000
REQUEST TIME			0.	0.	0.	0.
HOLD TIME	0		0.	0.	0.	0.
INTO BUSY	48					
OUT OF BUSY	48					
# BUSY		0.	0.004	0.066	0.	1.000
BUSY TIME			325.515	0.051	325.500	325.750
# INACTIVE		0.	0.	0.	0.	0.
INTO WAIT	48					
OUT OF WAIT	48					
# WAITING		0.	0.	0.	0.	1.000
WAIT TIME			0.	0.	0.	0.

CURRENTLY ALLOCATED
TO PROCESSES: NONE

PROCESSES CURRENTLY
WAITING: NONE

RESOURCE	TOTAL NUMBER	CURRENT...	MEAN......	STD DEV...	MINIMUM...	MAXIMUM...
CH7_B						
# IDLE		1.000	0.949	0.220	0.	1.000
REQUEST TIME			0.	0.	0.	0.
HOLD TIME	0		0.	0.	0.	0.
INTO BUSY	166					
OUT OF BUSY	166					
# BUSY		0.	0.051	0.220	0.	1.000
BUSY TIME			1103.069	184.486	813.750	1220.750
# INACTIVE		0.	0.	0.	0.	0.
INTO WAIT	166					
OUT OF WAIT	166					
# WAITING		0.	0.	0.	0.	1.000
WAIT TIME			0.	0.	0.	0.

CURRENTLY ALLOCATED
TO PROCESSES: NONE

PROCESSES CURRENTLY
WAITING: NONE

RESOURCE	TOTAL NUMBER	CURRENT...	MEAN......	STD DEV...	MINIMUM...	MAXIMUM...
CH8_A						
# IDLE		1.000	0.962	0.191	0.	1.000
REQUEST TIME			768.898	1047.365	0.	4582.375
HOLD TIME	0		0.	0.	0.	0.
INTO BUSY	168					
OUT OF BUSY	168					
# BUSY		0.	0.038	0.191	0.	1.000
BUSY TIME			813.802	0.054	813.750	814.000
# INACTIVE		0.	0.	0.	0.	0.
INTO WAIT	168					
OUT OF WAIT	168					
# WAITING		0.	0.036	0.312	0.	6.000
WAIT TIME			768.898	1047.365	0.	4582.375

PAGE 30

CURRENTLY ALLOCATED
TO PROCESSES: NONE

PROCESSES CURRENTLY
WAITING: NONE

RESOURCE	TOTAL NUMBER	CURRENT...	MEAN......	STD DEV...	MINIMUM...	MAXIMUM...
CH8_B						
# IDLE		1.000	0.945	0.228	0.	1.000
REQUEST TIME			0.	0.	0.	0.
HOLD TIME	0		0.	0.	0.	0.
INTO BUSY	286					
OUT OF BUSY	286					
# BUSY		0.	0.055	0.228	0.	1.000
BUSY TIME			694.874	440.708	325.500	1220.750
# INACTIVE		0.	0.	0.	0.	0.
INTO WAIT	286					
OUT OF WAIT	286					
# WAITING		0.	0.	0.	0.	1.000
WAIT TIME			0.	0.	0.	0.

CURRENTLY ALLOCATED
TO PROCESSES: NONE

PROCESSES CURRENTLY
WAITING: NONE

RESOURCE	TOTAL NUMBER	CURRENT...	MEAN......	STD DEV...	MINIMUM...	MAXIMUM...
CH9_A						
# IDLE		1.000	1.000	0.	1.000	1.000
REQUEST TIME			0.	0.	0.	0.
HOLD TIME	0		0.	0.	0.	0.
INTO BUSY	0					
OUT OF BUSY	0					
# BUSY		0.	0.	0.	0.	0.
BUSY TIME			0.	0.	0.	0.
# INACTIVE		0.	0.	0.	0.	0.
INTO WAIT	0					
OUT OF WAIT	0					

```
     # WAITING                   0.         0.         0.         0.         0.
     WAIT TIME                              0.         0.         0.         0.

    CURRENTLY ALLOCATED
           TO PROCESSES: NONE

    PROCESSES CURRENTLY
                WAITING: NONE

                      TOTAL
RESOURCE             NUMBER   CURRENT... MEAN...... STD DEV... MINIMUM... MAXIMUM...
=============== ======== ========== ========== ========== ========== ==========
CH9_B
         # IDLE                  1.000      1.000      0.         1.000      1.000
   REQUEST TIME                             0.         0.         0.         0.
      HOLD TIME            0                0.         0.         0.         0.

      INTO BUSY            0
    OUT OF BUSY            0
         # BUSY                  0.         0.         0.         0.         0.
      BUSY TIME                             0.         0.         0.         0.

     # INACTIVE                  0.         0.         0.         0.         0.

      INTO WAIT            0
    OUT OF WAIT            0
      # WAITING                  0.         0.         0.         0.         0.
      WAIT TIME                             0.         0.         0.         0.

    CURRENTLY ALLOCATED
           TO PROCESSES: NONE

    PROCESSES CURRENTLY
                WAITING: NONE

                      TOTAL
RESOURCE             NUMBER   CURRENT... MEAN...... STD DEV... MINIMUM... MAXIMUM...
=============== ======== ========== ========== ========== ========== ==========
CHQ
         # IDLE                  1.000      0.726      0.446      0.         1.000
   REQUEST TIME                           929.313   1924.186      0.     12569.750
      HOLD TIME            0                0.         0.         0.         0.

      INTO BUSY          827
    OUT OF BUSY          827
         # BUSY                  0.         0.274      0.446      0.         1.000
      BUSY TIME                          1192.101   2058.180      0.      6000.000

     # INACTIVE                  0.         0.         0.         0.         0.
```

INTO WAIT	827					
OUT OF WAIT	827					
# WAITING		0.	0.213	0.653	0.	7.000
WAIT TIME			929.313	1924.185	0.	12569.750

CURRENTLY ALLOCATED
TO PROCESSES: NONE

PROCESSES CURRENTLY
WAITING: NONE

RESOURCE	TOTAL NUMBER	CURRENT...	MEAN......	STD DEV...	MINIMUM...	MAXIMUM...
DK1						
# IDLE		1.000	0.998	0.045	0.	1.000
REQUEST TIME			0.	0.	0.	0.
HOLD TIME	0		0.	0.	0.	0.
INTO BUSY	168					
OUT OF BUSY	168					
# BUSY		0.	0.002	0.045	0.	1.000
BUSY TIME			44.071	10.937	3.656	85.500
# INACTIVE		0.	0.	0.	0.	0.
INTO WAIT	168					
OUT OF WAIT	168					
# WAITING		0.	0.	0.	0.	1.000
WAIT TIME			0.	0.	0.	0.

CURRENTLY ALLOCATED
TO PROCESSES: NONE

PROCESSES CURRENTLY
WAITING: NONE

RESOURCE	TOTAL NUMBER	CURRENT...	MEAN......	STD DEV...	MINIMUM...	MAXIMUM...
HQ						
# IDLE		1.000	0.728	0.445	0.	1.000
REQUEST TIME			1073.888	2372.956	0.	15611.250
HOLD TIME	0		0.	0.	0.	0.
INTO BUSY	410					
OUT OF BUSY	410					
# BUSY		0.	0.272	0.445	0.	1.000

```
PAGE   33
    BUSY TIME                               2385.677   2615.805        0.      6000.063

  # INACTIVE                         0.         0.         0.        0.          0.

    INTO WAIT         410
  OUT OF WAIT         410
    # WAITING                        0.         0.122      0.380     0.          3.000
    WAIT TIME                               1073.888   2372.956      0.      15611.250

  CURRENTLY ALLOCATED
         TO PROCESSES: NONE

  PROCESSES CURRENTLY
              WAITING: NONE

                    TOTAL
RESOURCE            NUMBER   CURRENT... MEAN...... STD DEV... MINIMUM... MAXIMUM...
=============== ======== ========== ========== ========== ========== ==========
L3
       # IDLE                     1.000      0.925      0.263      0.          1.000
    REQUEST TIME                            69.511    341.462      0.       2224.750
       HOLD TIME          0                  0.         0.         0.          0.

    INTO BUSY          96
  OUT OF BUSY          96
       # BUSY                        0.         0.075      0.263     0.          1.000
    BUSY TIME                               2800.000   1200.000   1600.000   4000.000

  # INACTIVE                         0.         0.         0.        0.          0.

    INTO WAIT          96
  OUT OF WAIT          96
    # WAITING                        0.         0.002      0.043     0.          1.000
    WAIT TIME                                 69.511    341.462      0.       2224.750

  CURRENTLY ALLOCATED
         TO PROCESSES: NONE

  PROCESSES CURRENTLY
              WAITING: NONE

                    TOTAL
RESOURCE            NUMBER   CURRENT... MEAN...... STD DEV... MINIMUM... MAXIMUM...
=============== ======== ========== ========== ========== ========== ==========
SW1
       # IDLE                     1.000      0.223      0.416      0.          1.000
    REQUEST TIME                           30346.677  21254.286      0.      84387.313
       HOLD TIME          0                  0.         0.         0.          0.
```

```
     INTO BUSY      594
   OUT OF BUSY      594
        # BUSY               0.          0.777      0.416      0.         1.000
     BUSY TIME                        4707.071   1745.716   1600.000   6000.000

    # INACTIVE               0.          0.         0.         0.         0.

     INTO WAIT      594
   OUT OF WAIT      594
     # WAITING               0.          5.007      5.432      0.        22.000
     WAIT TIME                       30346.677  21254.285      0.     84387.313

   CURRENTLY ALLOCATED
          TO PROCESSES: NONE

   PROCESSES CURRENTLY
              WAITING: NONE

                 TOTAL
RESOURCE         NUMBER   CURRENT... MEAN...... STD DEV... MINIMUM... MAXIMUM...
=============== ======== ========== ========== ========== ========== ==========
SW2
        # IDLE            1.000       0.729      0.445      0.         1.000
  REQUEST TIME                        373.714   1174.642      0.      7902.750
     HOLD TIME         0                0.         0.         0.         0.

     INTO BUSY      214
   OUT OF BUSY      214
        # BUSY               0.          0.271      0.445      0.         1.000
     BUSY TIME                        4564.486   1782.954   1600.000   6000.000

    # INACTIVE               0.          0.         0.         0.         0.

     INTO WAIT      214
   OUT OF WAIT      214
     # WAITING               0.          0.022      0.160      0.         2.000
     WAIT TIME                         373.714   1174.642      0.      7902.750

   CURRENTLY ALLOCATED
          TO PROCESSES: NONE

   PROCESSES CURRENTLY
              WAITING: NONE

                 TOTAL
RESOURCE         NUMBER   CURRENT... MEAN...... STD DEV... MINIMUM... MAXIMUM...
=============== ======== ========== ========== ========== ========== ==========
SW3
        # IDLE            1.000       0.542      0.498      0.         1.000
```

PAGE 35

REQUEST TIME			3857.755	5603.183	0.	24434.500
HOLD TIME	0		0.	0.	0.	0.
INTO BUSY	454					
OUT OF BUSY	454					
# BUSY		0.	0.458	0.498	0.	1.000
BUSY TIME			3631.718	1742.260	1599.992	6000.000
# INACTIVE		0.	0.	0.	0.	0.
INTO WAIT	454					
OUT OF WAIT	454					
# WAITING		0.	0.487	1.150	0.	7.000
WAIT TIME			3857.755	5603.183	0.	24434.500

CURRENTLY ALLOCATED
TO PROCESSES: NONE

PROCESSES CURRENTLY
WAITING: NONE

```
SIMULATION TIME =  3600000.00000 US

ACTION REPORT

                   TOTAL                                                % TIME
ACTION             SAMPLES  MEAN.....  STD DEV...  MINIMUM..  MAXIMUM...  OF TOTAL.
=============== ======= ========= ========= ========= ========= ========
FORMAT
    USEFUL TIME      168      2.000       0.          2.000       2.000      0.009
     DELAY TIME      168   1523.546    2269.177       0.       9663.500
    WASTED TIME        0      0.          0.          0.          0.

                   TOTAL                                                % TIME
ACTION             SAMPLES  MEAN.....  STD DEV...  MINIMUM..  MAXIMUM...  OF TOTAL.
=============== ======= ========= ========= ========= ========= ========
LATENCY
    USEFUL TIME      168     14.678       8.537       0.063      29.938      0.068
     DELAY TIME      168      0.          0.          0.          0.
    WASTED TIME        0      0.          0.          0.          0.

                   TOTAL                                                % TIME
ACTION             SAMPLES  MEAN.....  STD DEV...  MINIMUM..  MAXIMUM...  OF TOTAL.
=============== ======= ========= ========= ========= ========= ========
ROUTE_OH
    USEFUL TIME     2570   4337.743    1876.129    1599.992    6000.063    309.667
     DELAY TIME     2570      0.          0.          0.          0.
    WASTED TIME        0      0.          0.          0.          0.

                   TOTAL                                                % TIME
ACTION             SAMPLES  MEAN.....  STD DEV...  MINIMUM..  MAXIMUM...  OF TOTAL.
=============== ======= ========= ========= ========= ========= ========
SEEK
    USEFUL TIME      168     29.374      17.214       0.188      59.859      0.137
     DELAY TIME      168      0.          0.          0.          0.
    WASTED TIME        0      0.          0.          0.          0.

                   TOTAL                                                % TIME
ACTION             SAMPLES  MEAN.....  STD DEV...  MINIMUM..  MAXIMUM...  OF TOTAL.
=============== ======= ========= ========= ========= ========= ========
UPDATE
    USEFUL TIME      354      7.522       5.272       0.         11.250      0.074
     DELAY TIME      354   1774.778    3092.568       0.      15611.250
    WASTED TIME        0      0.          0.          0.          0.

                   TOTAL                                                % TIME
ACTION             SAMPLES  MEAN.....  STD DEV...  MINIMUM..  MAXIMUM...  OF TOTAL.
=============== ======= ========= ========= ========= ========= ========
XFER
```

USEFUL TIME	168	0.019	0.037	0.	0.250	+0.902E-04
DELAY TIME	168	0.	0.	0.	0.	
WASTED TIME	0	0.	0.	0.	0.	

ACTION	TOTAL SAMPLES	MEAN.....	STD DEV...	MINIMUM..	MAXIMUM...	% TIME OF TOTAL.
XFER OH						
USEFUL TIME	2048	694.152	438.196	81.250	1220.750	39.490
DELAY TIME	2048	0.	0.	0.	0.	
WASTED TIME	0	0.	0.	0.	0.	

```
SIMULATION TIME =    3600000.00000 US

PROCESS REPORT

                    TOTAL
PROCESS            SAMPLES. SUM....... MEAN...... STD DEV... MINIMUM... MAXIMUM...
=============== ======== ========== ========== ========== ========== ==========
ABUPDATE
           TOTAL      118      7.946      0.067      0.059      0.         0.126
    PROCESS WAIT        0      0.         0.         0.         0.         0.
   RESOURCE WAIT        0      0.         0.         0.         0.         0.

        TOTAL #  # AUTO   # CALL   # OF     # NOT    # TIMES
        SCHEDULE SCHEDULE SCHEDULE COMPLETE COMPLETE SUSPEND.
        ======== ======== ======== ======== ======== ========
             118        0      118      118        0        0

         ITEM     CREATED  RECEIVED SENT     DESTR'D
         ======== ======== ======== ======== ========
         MSG             0        0        0        0

                   PROCESS  HOLDING  TIME
         ITEM     # SMPLS  MEAN...... MINIMUM.. MAXIMUM... STD DEV...
         ======== ======== ========== ========== ========== ==========
         MSG           118       0.07      0.          0.13       0.06

PROCESS            DESCRIPTION
======== ======== =====================================================
ABUPDATE           UPDATE DATA FROM AIRBASE

COUNT ENTRY     OPCODE   PARM     PARM     PARM     COMMENT
===== ======== ======== ======== ======== ======== ==========================
  118           START             NO
  118           GIVEN    MSG
  118           RETURN   MSG
  118           UPDATE   CONSTANT 0.1               TIME CONSUMED IN UPDATING
  118                    US       RESUME
  118           END

                    TOTAL
PROCESS            SAMPLES. SUM....... MEAN...... STD DEV... MINIMUM... MAXIMUM...
=============== ======== ========== ========== ========== ========== ==========
AB_DATA
           TOTAL      118      0.         0.         0.         0.         0.
    PROCESS WAIT        0      0.         0.         0.         0.         0.
   RESOURCE WAIT        0      0.         0.         0.         0.         0.

        TOTAL #  # AUTO   # CALL   # OF     # NOT    # TIMES
```

```
         SCHEDULE SCHEDULE SCHEDULE COMPLETE COMPLETE SUSPEND.
         ======== ======== ======== ======== ======== ========
              118      118        0      118        0        0

PROCESS           DESCRIPTION
======== ======== ==================================================
AB_DATA           AIR BASE STATUS BROUADCAST TO ALL OTHER NODES

COUNT ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
===== ======== ======== ======== ======== ======== ==========================
  118          START             NO
  118          GIVEN    MSG
  118          RETURN   MSG
  118          CALL     MRS      NOWAIT   10       PROCESS REQUEST TO CHQ
  118          GIVEN    CHQ_DATA 10       $REQNORE
  118                   750      CHQ      MSG
  118          CALL     MRS      NOWAIT   10       PROCESS REQUEST TO HQ
  118          GIVEN    HQ_DATA  10       $REQNORE
  118                   750      HQ       MSG
  118          ASSIGN   $CNODE                     CURRENT NODE
  118                   CNODE
  118          COMPARE  CNODE             EQ       TEST FOR CURRENT NODE
  118                   AB1               AB1
   59          CALL     MRS      NOWAIT   10       PROCESS REQUEST TO AB1
   59          GIVEN    ABUPDATE 10       $REQNORE
   59                   750      AB1      MSG
   59          BRANCH   END      100               BRANCH TO THE END
   59 AB1      ENTRY                               ENTRY FROM COMPARE NODE
   59          CALL     MRS      NOWAIT   10       PROCESS REQUEST TO AB2
   59          GIVEN    ABUPDATE 10       $REQNORE
   59                   750      AB2      MSG
  118 END      ENTRY                               ENDTRY FROM REQUEST TO A1H
  118          END

                         TOTAL
PROCESS                SAMPLES  SUM.......  MEAN......  STD DEV...  MINIMUM...  MAXIMUM...
============== ======== ========== ========== ========== ========== ==========
AB_REQ
         TOTAL        120 +1.505E+07 125424.735  36615.237  22605.588 194855.438
  PROCESS WAIT        120 +1.505E+07 125424.735  36615.237  22605.588 194855.438
 RESOURCE WAIT          0         0.         0.         0.         0.         0.

         TOTAL #  # AUTO   # CALL   # OF     # NOT    # TIMES
         SCHEDULE SCHEDULE SCHEDULE COMPLETE COMPLETE SUSPEND.
         ======== ======== ======== ======== ======== ========
              120      120        0      120        0        0

PROCESS           DESCRIPTION
======== ======== ==================================================
```

```
AB_REQ             AIRBASE REQUEST FOR PLANS REPORT FROM CHQ

COUNT ENTRY     OPCODE   PARM     PARM     PARM     COMMENT
===== ======== ======== ======== ======== ======== ========================
  120           START             NO
  120           GIVEN    MSG
  120           RETURN   MSG
  120           CALL     MRS      WAIT     5        PROCESS REQUEST TO CHQ
  120           GIVEN    PLANS    5        $REQRESP
  120                    200      CHQ      MSG
  120           END

                  TOTAL
PROCESS          SAMPLES. SUM....... MEAN...... STD DEV... MINIMUM... MAXIMUM...
=============== ======== ========== ========== ========== ========== ==========
CHANPROC
          TOTAL     2048 +7.946E+07  38798.344  33437.938   4813.750 183450.813
   PROCESS WAIT     2048 +7.791E+07  38041.118  33523.332   4000.000 183125.313
  RESOURCE WAIT     2048 129174.794     63.074    366.749         0.   4582.375

        TOTAL #  # AUTO   # CALL   # OF     # NOT    # TIMES
        SCHEDULE SCHEDULE SCHEDULE COMPLETE COMPLETE SUSPEND.
        ======== ======== ======== ======== ======== ========
           2048        0     2048     2048        0        0

         ITEM     CREATED  RECEIVED SENT     DESTR'D
         ======== ======== ======== ======== ========
         MSG             0        0        0        0

                   PROCESS  HOLDING  TIME
         ITEM     # SMPLS  MEAN...... MINIMUM..  MAXIMUM... STD DEV...
         ======== ======== ========== ========== ========== ==========
         MSG          2048     757.23      81.25    5396.19     584.48

PROCESS           DESCRIPTION
======== ======== ====================================================
CHANPROC          FULL AND HALF DUPLEX CHANNEL LOGIC

COUNT ENTRY     OPCODE   PARM     PARM     PARM     COMMENT
===== ======== ======== ======== ======== ======== ========================
 2048           START    ALL      NO
 2048           GIVEN    MSG
 2048           ASSIGN   MSG      CNODE             SET INTERNAL NODE CURRENT
 2048                    $CNODE
 2048           ASSIGN   MSG      TNODE             GET DESTINATION NODE
 2048                    TO_NODE
 2048           ASSIGN   $NXTNODE TO_NODE           SET NEXT NODE TO DESTN
 2048                    NXT_NODE
 2048           ASSIGN   $CHANNEL TO_NODE           GET CHANNEL TO NEXT NODE
```

```
 2048                      CHANNEL
 2048          ALLOC       CHANNEL  1          ALL       OBTAIN CHANNEL FOR XFER
 2048                      $PRIORTY
 2048          ASSIGN      CHANNEL  RATE                 WHAT IS CHANNEL RATE?
 2048                      VSPEED
 2048          ASSIGN      MSG      LENGTH               MESSAGE LENGTH
 2048                      VLENGTH
 2048          EVAL        VM_OVHD                       CALCULATE TRANSFER TIME
 2048                      VSPEED*VLENGTH
 2048          XFER_OH     CONSTANT VM_OVHD              DELAY DUE TO TRANSFER TIME
 2048                      US       RESUME
 2048          ASSIGN      NXT_NODE                      MSG RESIDES IN NEXT NODE
 2048                      MSG_     CNODE
 2048          ASSIGN      NXT_NODE                      SET INTERNAL NODE REGISTER
 2048                      $CNODE
 2048          DEALLOC     CHANNEL  1                    FREE UP CHANNEL AFTER XFER
 2048          CALL        NODEPROC WAIT       0         ROUTE MESSAGE TO NEXT NODE
 2048          GIVEN       MSG
 2048          END

                  TOTAL
PROCESS         SAMPLES. SUM....... MEAN...... STD DEV... MINIMUM... MAXIMUM...
=============== ======== ========== ========== ========== ========== ==========
CHQ_DATA
          TOTAL      118 335577.578   2843.878   3260.056     11.250  12581.000
   PROCESS WAIT        0         0.         0.         0.         0.         0.
  RESOURCE WAIT      103 334250.078   3245.146   3284.194     11.250  12569.750

         TOTAL #  # AUTO   # CALL   # OF     # NOT    # TIMES
         SCHEDULE SCHEDULE SCHEDULE COMPLETE COMPLETE SUSPEND.
         ======== ======== ======== ======== ======== ========
              118        0      118      118        0      103

         ITEM     CREATED  RECEIVED SENT     DESTR'D
         ======== ======== ======== ======== ========
         MSG             0        0        0        0

                   PROCESS  HOLDING  TIME
         ITEM     # SMPLS  MEAN...... MINIMUM..  MAXIMUM... STD DEV...
         ======== ======== ========== ========== ========== ==========
         MSG           118    2843.88      11.25   12581.00    3260.06

PROCESS           DESCRIPTION
======== ======== ==================================================
CHQ_DATA          CHQ GETS MESSAGE, FORMULATES RESPONSE, AND REPLIES

COUNT ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
===== ======== ======== ======== ======== ======== ========================
  118          START             NO
```

```
PAGE   42
  118          GIVEN    MSG
  118          RETURN   MSG
  118          ASSIGN   MSG       LENGTH              MAKE MSG-LENGTH = V_LENGTH
  118                   V_LENGTH
  118          EVAL     V_TIME                        EVALUATE MSG PROCESS TIME
  118                   .016*V_LENGTH
  205          UPDATE   CONSTANT V_TIME               PROCESSING TIME CONSUMED
  205                   US        RESUME
  118          END

                  TOTAL
PROCESS          SAMPLES. SUM........ MEAN...... STD DEV... MINIMUM... MAXIMUM...
=============== ======== ========== ========== ========== ========== ==========
DESTPROC
          TOTAL      690 +1.014E+07  14697.168  25948.783         0.  108088.125
   PROCESS WAIT      690 +9.725E+06  14094.377  25102.148         0.  106124.188
  RESOURCE WAIT      522 415925.555    796.792   2890.029         0.   22000.008

        TOTAL #  # AUTO   # CALL   # OF     # NOT    # TIMES
        SCHEDULE SCHEDULE SCHEDULE COMPLETE COMPLETE SUSPEND.
        ======== ======== ======== ======== ======== ========
             690        0      690      690        0        0

         ITEM     CREATED  RECEIVED SENT     DESTR'D
         ======== ======== ======== ======== ========
         MSG             0        0        0      522

                  PROCESS  HOLDING  TIME
         ITEM     # SMPLS  MEAN...... MINIMUM..  MAXIMUM... STD DEV...
         ======== ======== ========== ========== ========== ==========
         MSG          1212     343.17         0.   22000.01    1937.25

PROCESS           DESCRIPTION
======== ======== ==========================================================
DESTPROC          PROCESSING AT DESTINATION OF MESSAGE

COUNT ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
===== ======== ======== ======== ======== ======== ========================
  690          START    ALL      NO
  690          GIVEN    MSG
  690          ASSIGN   MSG      CNODE             CURRENT NODE
  690                   C_NODE
  690          COMPARE  MSG      TYPE     EQ       IF RESPONSE,D DESTROY
  690                   $RESP             DESTROY
  522          ALLOC    C_NODE   1        ALL      ALLOCATE CURRENT NODE
  522                   $PRIORTY
  522          ASSIGN   MSG      RPROC             EXECUTE THE CALLED PROCESS
  522                   PROCESS
  522          ASSIGN   MSG      RPROCPRI          SET PRIORITY FOR REQ PROC
```

```
 522                     PRIORITY
 522           CALL      PROCESS  WAIT      PRIORITY WAIT UNTIL COMPLETE
 522           GIVEN     MSG
 522           RETURN    MSG
 522           DEALLOC   C_NODE   1                  DEALLOCATE CURRENT NODE
 522           COMPARE   MSG      TYPE      EQ       NO RESPONSE REQ -> DESTROY
 522                     $REQNORE           DESTROY
 168           ASSIGN    $RESP                       CHANGE MSG RESPONSE TYPE
 168                     MSG      TYPE
 168           ASSIGN    MSG      FNODE              SWITCH FROM AND TO NODES
 168                     MSG      TNODE
 168           ASSIGN    MSG      CNODE              CURRENT NODE IS FROM NODE
 168                     MSG      FNODE
 168           CALL      CHANPROC WAIT      0        RETURN MESSAGE TO ORIGIN
 168           GIVEN     MSG
 168           BRANCH    END      100
 522 DESTROY   ENTRY                                 TERMINATE MESSAGE AT DEST
 522           DESTROY   MSG                         TERMINATE MSG
 690 END       ENTRY
 690           END
```

PROCESS	TOTAL SAMPLES.	SUM.......	MEAN......	STD DEV...	MINIMUM...	MAXIMUM...
DISK_OP						
TOTAL	168	7403.990	44.071	18.937	3.656	85.500
PROCESS WAIT	0	0.	0.	0.	0.	0.
RESOURCE WAIT	168	0.	0.	0.	0.	0.

TOTAL # SCHEDULE	# AUTO SCHEDULE	# CALL SCHEDULE	# OF COMPLETE	# NOT COMPLETE	# TIMES SUSPEND.
168	0	168	168	0	0

PROCESS	DESCRIPTION
DISK_OP	OPERATION OF DISK

```
COUNT ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
===== ======== ======== ======== ======== ======== ========================
 168           START             NO
 168           GIVEN    LENGTH   DISK
 168           ASSIGN   DISK     SPEED             MAKE DISK SPEED = V_SPEED
 168                    V_SPEED
 168           EVAL     XFERTIME                   TRANSFER TIME CALCULATED
 168                    LENGTH/V_SPEED
 168           ALLOC    DISK     1        PARTIAL  DISK ALLOCATED
 168                    $PRIORTY
 168           ASSIGN   DISK     SEEK              MAKE SEEKTIME = SEEK
```

```
PAGE    44
  168                 SEEKTIME
  168         SEEK     UNIFORM  SEEKTIME SEEKTIME TIME FOR SEEK IS CONSUMED•
  168                  US       RESUME
  168         ASSIGN   DISK     LATENCY           MAKE DISK LATENCY=LATETIME
  168                  LATETIME
  168         LATENCY  UNIFORM  LATETIME LATETIME TIME CONSUMED FOR LATENCY
  168                  US       RESUME
  168         XFER     CONSTANT XFERTIME          TRANSFER TIME CONSUMED
  168                  US       RESUME
  168         DEALLOC  DISK     1                 DISK RESOURCE DEALLOCATED
  168         END

                TOTAL
PROCESS        SAMPLES. SUM.......  MEAN......  STD DEV...  MINIMUM...  MAXIMUM...
============== ======== ==========  ==========  ==========  ==========  ==========
HQ_DATA
        TOTAL       118 318233.750    2696.896    3636.755      11.250   15622.500
  PROCESS WAIT        Ø         Ø.          Ø.          Ø.          Ø.          Ø.
 RESOURCE WAIT       78 316906.250    4062.901    3788.903      11.250   15611.250

        TOTAL #  # AUTO   # CALL   # OF     # NOT    # TIMES
        SCHEDULE SCHEDULE SCHEDULE COMPLETE COMPLETE SUSPEND.
        ======== ======== ======== ======== ======== ========
           118        Ø      118      118        Ø       78

         ITEM     CREATED  RECEIVED SENT     DESTR'D
         ======== ======== ======== ======== ========
         MSG             Ø        Ø        Ø        Ø

                  PROCESS  HOLDING  TIME
         ITEM     # SMPLS  MEAN......  MINIMUM..  MAXIMUM...  STD DEV...
         ======== ======== ==========  ========== ==========  ==========
         MSG           118    2696.90       11.25   15622.50     3636.76

PROCESS           DESCRIPTION
======== ======== ==================================================
HQ_DATA           HQ GETS MESSAGE, FORMULATES RESPONSE, AND REPLIES

COUNT ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
===== ======== ======== ======== ======== ======== ========================
  118          START             NO
  118          GIVEN    MSG
  118          RETURN   MSG
  118          ASSIGN   MSG      LENGTH            MAKE MSG_LENGTH = V_LENGTH
  118                   V_LENGTH
  118          EVAL     V_TIME                     EVALUATE MSG PROCESS TIME
  118                   .Ø15•V_LENGTH
  196          UPDATE   CONSTANT V_TIME            PROCESSING TIME CONSUMED
  196                   US       RESUME
```

```
PAGE   45
 118          END

                    TOTAL
PROCESS             SAMPLES. SUM....... MEAN...... STD DEV... MINIMUM... MAXIMUM...
=============== ======== ========== ========== ========== ========== ==========
HQ_REQ
          TOTAL          48 +1.852E+06  38585.016   9495.574  27754.250  63392.188
   PROCESS WAIT          48 +1.852E+06  38585.016   9495.574  27754.250  63392.188
  RESOURCE WAIT           0         0.         0.         0.         0.         0.

        TOTAL #  # AUTO   # CALL   # OF     # NOT    # TIMES
        SCHEDULE SCHEDULE SCHEDULE COMPLETE COMPLETE SUSPEND.
        ======== ======== ======== ======== ======== ========
              48       48        0       48        0        0

PROCESS             DESCRIPTION
======== ======== ====================================================
HQ_REQ              HQ REQUEST FOR STATUS DISPLAY FROM CHQ

COUNT ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
===== ======== ======== ======== ======== ======== ==========================
   48          START    L3       NO
   48          GIVEN    MSG
   48          RETURN   MSG
   48          CALL     MRS      WAIT     4        MAKES I/O REQUEST TO CHQ
   48          GIVEN    PLANS    4        $REQRESP
   48                   200      CHQ      MSG
   48          END

                    TOTAL
PROCESS             SAMPLES. SUM....... MEAN...... STD DEV... MINIMUM... MAXIMUM...
=============== ======== ========== ========== ========== ========== ==========
MRS
          TOTAL         522 +3.753E+07  71890.936  38971.351  22605.588 194855.438
   PROCESS WAIT         522 +3.753E+07  71890.936  38971.351  22605.588 194855.438
  RESOURCE WAIT           0         0.         0.         0.         0.         0.

        TOTAL #  # AUTO   # CALL   # OF     # NOT    # TIMES
        SCHEDULE SCHEDULE SCHEDULE COMPLETE COMPLETE SUSPEND.
        ======== ======== ======== ======== ======== ========
             522        0      522      522        0        0

         ITEM     CREATED  RECEIVED SENT     DESTR'D
         ======== ======== ======== ======== ========
         MSG           522        0        0        0

                  PROCESS  HOLDING  TIME
         ITEM     # SMPLS  MEAN...... MINIMUM..  MAXIMUM... STD DEV...
         ======== ======== ========== ========== ========== ==========
```

```
         MSG           522       0.         0.         0.         0.

PROCESS           DESCRIPTION
======== ======== ====================================================
MRS               GENERATE A PROCESS REQUEST MESSAGE AND INITIATE I/O

COUNT ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
===== ======== ======== ======== ======== ======== ========================
  522          START    ALL      NO
  522          GIVEN    PROCESS  PRIORITY MSG_TYPE
  522                   MSG_LNTH TO_NODE  MSG
  522          CREATE   MSG                        CREATE MESSAGE
  522          ASSIGN   MSG_LNTH                   SET MESSAGE LENGTH
  522                   MSG      LENGTH
  522          ASSIGN   PROCESS                    SET PROCESS
  522                   MSG      RPROC
  522          ASSIGN   PRIORITY                   SET PRIORITY
  522                   MSG      RPROCPRI
  522          ASSIGN   TO_NODE                    SET DESTINATION
  522                   MSG      TNODE
  522          ASSIGN   MSG_TYPE                   SET MESSAGE TYPE
  522                   MSG      TYPE
  522          CALL     NODEPROC WAIT     0        EXECUTIVE SERVICING OF MSG
  522          GIVEN    MSG
  522          END

                 TOTAL
PROCESS         SAMPLES. SUM....... MEAN...... STD DEV... MINIMUM... MAXIMUM...
=============== ======== ========== ========== ========== ========== ==========
NODEPROC
          TOTAL     2570 +1.164E+08  44916.450  37275.894   4000.000 194866.438
  PROCESS WAIT      2570 +8.080E+07  31439.400  33476.749         0. 183450.813
 RESOURCE WAIT      2570 +2.349E+07   9139.307  16030.260         0.  84387.313

        TOTAL #  # AUTO   # CALL   # OF     # NOT    # TIMES
        SCHEDULE SCHEDULE SCHEDULE COMPLETE COMPLETE SUSPEND.
        ======== ======== ======== ======== ======== ========
            2570        0     2570     2570        0        0

         ITEM     CREATED  RECEIVED SENT     DESTR'D
         ======== ======== ======== ======== ========
         MSG             0        0        0        0

                   PROCESS  HOLDING  TIME
         ITEM     # SMPLS  MEAN...... MINIMUM..  MAXIMUM... STD DEV...
         ======== ======== ========== ========== ========== ==========
         MSG          2570   13477.05    1599.99   85987.31   16095.00

PROCESS           DESCRIPTION
```

NODEPROC NODAL PROCESSING AND ROUTING

COUNT	ENTRY	OPCODE	PARM	PARM	PARM	COMMENT
2570		START	ALL	NO		
2570		GIVEN	MSG			
2570		ASSIGN	MSG	CNODE		INDICATE CURRENT NODE
2570			C_NODE			
2570		ASSIGN	C_NODE	M_ROUTE		PROCESSING RATE OF NODE
2570			RT_OVHD			
2570		ASSIGN	MSG	LENGTH		GET MESSAGE LENGTH
2570			MSG_LNTH			
2570		EVAL	OVERHEAD			COMPUTE PROCESSING DELAY
2570			MSG_LNTH*RT_OVHD			
2570		ALLOC	C_NODE	1	ALL	ALLOCATE CURRENT NODE
2570			$PRIORTY			
2570		ROUTE_OH	CONSTANT	OVERHEAD		DELAY FOR ROUTING
2570			US	RESUME		
2570		DEALLOC	C_NODE	1		RELEASE C_NODE TO OTHERS
2570		COMPARE	MSG	CNODE	EQ	IS MSG AT DESTINATION?
2570			MSG	TNODE	CONTROL	
1880		CALL	CHANPROC	WAIT	0	FORWARD MSG TO CHANNEL
1880		GIVEN	MSG			
1880		BRANCH	END	100		
690	CONTROL	ENTRY				MESSAGE AT DESTINATION
690		CALL	DESTPROC	WAIT	0	CONTEXT SWITCH MESSAGE
690		GIVEN	MSG			
2570	END	ENTRY				
2570		END				

PROCESS	TOTAL SAMPLES.	SUM.......	MEAN......	STD DEV...	MINIMUM...	MAXIMUM...
PLANS						
TOTAL	168	270504.811	1610.148	2264.765	9.000	9723.125
PROCESS WAIT	168	7403.990	44.071	18.937	3.656	85.500
RESOURCE WAIT	152	262764.820	1728.716	1645.353	11.250	8180.375

TOTAL # SCHEDULE	# AUTO SCHEDULE	# CALL SCHEDULE	# OF COMPLETE	# NOT COMPLETE	# TIMES SUSPEND.
168	0	168	168	0	152

ITEM	CREATED	RECEIVED	SENT	DESTR'D
MSG	0	0	0	0

PROCESS HOLDING TIME

```
PAGE    48
          ITEM     # SMPLS  MEAN......  MINIMUM..  MAXIMUM...  STD DEV...
          ======== ======== ========== ========== ========== ==========
          MSG          168    1610.15       9.00    9723.13    2264.76

PROCESS           DESCRIPTION
======== ======== ====================================================
PLANS             REQUEST FOR PLANS FROM CHQ

COUNT ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
===== ======== ======== ======== ======== ======== =========================
  168          START    CHQ      NO
  168          GIVEN    MSG
  168          RETURN   MSG
  168          ASSIGN   MSG      LENGTH            MAKE MSG LENGTH = V_LENGTH
  168                   V_LENGTH
  168          EVAL     V_TIME                     EVALUATE MSG PROCESS TIME
  168                   .01*V_LENGTH
  314          FORMAT   CONSTANT V_TIME            TIME USED TO FORMAT PLANS
  314                   US       RESUME
  168          CALL     DISK_OP  WAIT     10       CALLING PROCESS DISK_OP
  168          GIVEN    500      DK1
  168          ASSIGN   500                        INCREASE MSG LENGTH
  168                   MSG      LENGTH
  168          END

                  TOTAL
PROCESS         SAMPLES. SUM.......  MEAN......  STD DEV...  MINIMUM...  MAXIMUM...
=============== ======== ========== ========== ========== ========== ==========
TRACE
          TOTAL        1        0.          0.          0.          0.          0.
   PROCESS WAIT        0        0.          0.          0.          0.          0.
  RESOURCE WAIT        0        0.          0.          0.          0.          0.

        TOTAL #  # AUTO   # CALL   # OF     # NOT    # TIMES
        SCHEDULE SCHEDULE SCHEDULE COMPLETE COMPLETE SUSPEND.
        ======== ======== ======== ======== ======== ========
               1        1        0        1        0        0

PROCESS           DESCRIPTION
======== ======== ====================================================
TRACE             TURN ON TRACE OUTPUT

COUNT ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
===== ======== ======== ======== ======== ======== =========================
    1          START    ALL      NO
    1          COMPARE  V_TRACE           EQ       TEST IF FLAG SET FOR TRACE
    1                   0                 NOTRACE
    0          TRACE    ON
    1 NOTRACE  ENTRY
```

PAGE 49
1
END

END
FILMED
MARCH, 1988
DTIC